Oracle® Fusion Middleware
XML Schema Reference for Oracle Business Intelligence Enterprise Edition
11g Release 1 (11.1.1)
E23950-04

July 2014
Describes how to use the Oracle BI Server MDS XML API and the legacy Oracle BI Server XML API, and provides reference information for each MDS and XML schema element.
Contents

Preface .................................................................................................................................................................. ix

Part I MDS XML Schema Elements

1 About the Oracle BI Server MDS XML API

1.1 About the Oracle BI Server MDS XML API....................................................................................... 1-1
1.2 Generating MDS XML......................................................................................................................... 1-2
1.2.1 Generating MDS XML from an Existing RPD Using the Administration Tool.......................... 1-2
1.2.2 Generating MDS XML from an Existing RPD Using a Command-Line Utility......................... 1-3
1.3 Executing MDS XML......................................................................................................................... 1-4
1.4 Generic MDS XML Elements and Attributes.................................................................................. 1-6
1.4.1 About MDS XML Object Identifiers.......................................................................................... 1-7
1.5 Example: Analysis of an MDS XML Logical Table Element.......................................................... 1-7
1.6 Topics of Interest in Other Guides.................................................................................................... 1-10
1.7 System Requirements and Certification.......................................................................................... 1-11

2 MDS XML Logical Elements

2.1 MDS XML Elements: Logical Column.............................................................................................. 2-1
2.2 MDS XML Elements: Business Model............................................................................................... 2-3
2.3 MDS XML Elements: Logical Complex Join..................................................................................... 2-4
2.4 MDS XML Elements: Logical Level................................................................................................... 2-5
2.5 MDS XML Elements: Dimension...................................................................................................... 2-7
2.6 MDS XML Elements: Logical Table.................................................................................................. 2-9
2.7 MDS XML Elements: Logical Foreign Key....................................................................................... 2-12
2.8 MDS XML Elements: Logical Key .................................................................................................... 2-13
2.9 MDS XML Elements: Logical Table Source.................................................................................... 2-14
2.9.1 Column Mapping ......................................................................................................................... 2-18
2.10 MDS XML Elements: Logical Source Folder............................................................................... 2-19
2.11 MDS XML Elements: Measure Definition..................................................................................... 2-20
2.12 MDS XML Elements: Aggregation Rule......................................................................................... 2-21
2.13 MDS XML Elements: Override Aggregation Rule......................................................................... 2-22
3 MDS XML Physical Elements

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>MDS XML Elements: Database</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>MDS XML Elements: Physical Catalog</td>
<td>3-4</td>
</tr>
<tr>
<td>3.3</td>
<td>MDS XML Elements: Physical Schema</td>
<td>3-4</td>
</tr>
<tr>
<td>3.4</td>
<td>MDS XML Elements: Physical Display Folder</td>
<td>3-5</td>
</tr>
<tr>
<td>3.5</td>
<td>MDS XML Elements: Physical Table</td>
<td>3-6</td>
</tr>
<tr>
<td>3.6</td>
<td>MDS XML Elements: Physical Column</td>
<td>3-11</td>
</tr>
<tr>
<td>3.7</td>
<td>MDS XML Elements: Connection Pool</td>
<td>3-12</td>
</tr>
<tr>
<td>3.8</td>
<td>MDS XML Elements: Physical Key</td>
<td>3-18</td>
</tr>
<tr>
<td>3.9</td>
<td>MDS XML Elements: Physical Foreign Key</td>
<td>3-19</td>
</tr>
<tr>
<td>3.10</td>
<td>MDS XML Elements: Complex Join</td>
<td>3-20</td>
</tr>
<tr>
<td>3.11</td>
<td>MDS XML Elements: Cube Table</td>
<td>3-21</td>
</tr>
<tr>
<td>3.12</td>
<td>MDS XML Elements: Physical Dimension</td>
<td>3-24</td>
</tr>
<tr>
<td>3.13</td>
<td>MDS XML Elements: Physical Hierarchy</td>
<td>3-26</td>
</tr>
<tr>
<td>3.14</td>
<td>MDS XML Elements: Physical Level</td>
<td>3-28</td>
</tr>
<tr>
<td>3.15</td>
<td>MDS XML Elements: Cube Column</td>
<td>3-29</td>
</tr>
<tr>
<td>3.16</td>
<td>MDS XML Elements: Analytic Workspace</td>
<td>3-31</td>
</tr>
<tr>
<td>3.17</td>
<td>MDS XML Elements: AW Cube</td>
<td>3-31</td>
</tr>
<tr>
<td>3.18</td>
<td>MDS XML Elements: AW Dimension</td>
<td>3-34</td>
</tr>
<tr>
<td>3.19</td>
<td>MDS XML Elements: AW Hierarchy</td>
<td>3-37</td>
</tr>
<tr>
<td>3.20</td>
<td>MDS XML Elements: AW Level</td>
<td>3-39</td>
</tr>
<tr>
<td>3.21</td>
<td>MDS XML Elements: AW Column</td>
<td>3-40</td>
</tr>
</tbody>
</table>

4 MDS XML Presentation Elements

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>MDS XML Elements: Presentation Catalog</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2</td>
<td>MDS XML Elements: Presentation Table</td>
<td>4-3</td>
</tr>
<tr>
<td>4.3</td>
<td>MDS XML Elements: Presentation Column</td>
<td>4-5</td>
</tr>
<tr>
<td>4.4</td>
<td>MDS XML Elements: Presentation Hierarchy</td>
<td>4-7</td>
</tr>
<tr>
<td>4.5</td>
<td>MDS XML Elements: Presentation Level</td>
<td>4-9</td>
</tr>
</tbody>
</table>

5 MDS XML Marketing, Security, and Other Elements

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>MDS XML Marketing Elements</td>
<td>5-1</td>
</tr>
<tr>
<td>5.1.1</td>
<td>MDS XML Elements: Catalog Link</td>
<td>5-1</td>
</tr>
<tr>
<td>5.1.2</td>
<td>MDS XML Elements: Target Level</td>
<td>5-2</td>
</tr>
<tr>
<td>5.1.3</td>
<td>MDS XML Elements: List Catalog</td>
<td>5-4</td>
</tr>
<tr>
<td>5.1.4</td>
<td>MDS XML Elements: Qualified List Item</td>
<td>5-4</td>
</tr>
<tr>
<td>5.1.5</td>
<td>MDS XML Elements: Qualifying Key</td>
<td>5-6</td>
</tr>
<tr>
<td>5.1.6</td>
<td>MDS XML Elements: Sampling Table</td>
<td>5-6</td>
</tr>
<tr>
<td>5.1.7</td>
<td>MDS XML Elements: Seg Catalog</td>
<td>5-7</td>
</tr>
<tr>
<td>5.2</td>
<td>MDS XML Security Elements</td>
<td>5-8</td>
</tr>
<tr>
<td>5.2.1</td>
<td>MDS XML Elements: Variable</td>
<td>5-8</td>
</tr>
<tr>
<td>5.2.2</td>
<td>MDS XML Elements: LDAP Server</td>
<td>5-9</td>
</tr>
<tr>
<td>5.2.3</td>
<td>MDS XML Elements: Init Block</td>
<td>5-11</td>
</tr>
<tr>
<td>5.2.4</td>
<td>MDS XML Elements: Custom Authenticator</td>
<td>5-14</td>
</tr>
</tbody>
</table>
Part II  XML Schema Elements

6  About the Oracle BI Server XML API
6.1  About the Oracle BI Server XML API ................................................................. 6-1
6.1.1  About Using the Oracle BI Server XML API to Merge and Append Objects .......... 6-2
6.2  Generating and Executing XML ............................................................................ 6-2
6.2.1  Generating XML from an Existing RPD ............................................................... 6-3
6.2.2  Executing XML in Offline Mode ......................................................................... 6-4
6.2.3  Executing XML Against the Oracle BI Server ..................................................... 6-5
6.3  Using the Oracle BI Server XML API .................................................................... 6-5
6.3.1  Maintaining Versions of the Metadata ................................................................. 6-6
6.3.2  Migrating Metadata from Another System ........................................................... 6-6
6.3.3  Applying a Metadata Patch Against a Running Oracle BI Server ....................... 6-7
6.3.4  Manipulating Metadata Programmatically .......................................................... 6-7
6.3.4.1  Example of Programmatically Creating a Repository Object Using XML ........ 6-7
6.3.5  Moving from Test to Production Environments .................................................... 6-10
6.3.6  Using XSL to Transform XML Files .................................................................... 6-12
6.3.7  Setting Permissions Using Command-Line Tools ................................................. 6-13
6.4  Generic XML Elements and Attributes .................................................................. 6-13
6.4.1  About Object Identifiers ..................................................................................... 6-14
6.4.2  About Naming Requirements for Repository Objects ......................................... 6-15

7  XML Logical Elements
7.1  XML Elements: Logical Column ............................................................................. 7-1
7.2  XML Elements: Business Model ............................................................................. 7-3
7.3  XML Elements: Logical Complex Join ................................................................. 7-4
7.4  XML Elements: Logical Level ................................................................................. 7-5
7.5  XML Elements: Dimension .................................................................................... 7-6
7.6  XML Elements: Logical Table ................................................................................ 7-8
7.7  XML Elements: Logical Foreign Key ...................................................................... 7-9
7.8  XML Elements: Logical Key ................................................................................... 7-11
7.9  XML Elements: Logical Table Source ................................................................. 7-12
7.9.1  Column Mapping ......................................................................................... 7-16
7.10 XML Elements: Logical Source Folder .............................................................. 7-17
7.11 XML Elements: Measure Definition ................................................................. 7-17
7.12 XML Elements: Aggregation Rule ....................................................................... 7-18
8 XML Physical Elements

8.1 XML Elements: Database ................................................................. 8-1
8.2 XML Elements: Physical Catalog ...................................................... 8-4
8.3 XML Elements: Physical Schema ..................................................... 8-4
8.4 XML Elements: Physical Display Folder ........................................... 8-5
8.5 XML Elements: Physical Table ......................................................... 8-5
8.6 XML Elements: Physical Column .................................................... 8-9
8.7 XML Elements: Connection Pool ..................................................... 8-11
8.8 XML Elements: Physical Key .......................................................... 8-16
8.9 XML Elements: Physical Foreign Key ............................................... 8-17
8.10 XML Elements: Complex Join ....................................................... 8-18
8.11 XML Elements: Cube Table ......................................................... 8-20
8.12 XML Elements: Physical Dimension .............................................. 8-23
8.13 XML Elements: Physical Hierarchy ............................................... 8-24
8.14 XML Elements: Physical Level ...................................................... 8-27
8.15 XML Elements: Cube Column ....................................................... 8-27
8.16 XML Elements: Analytic Workspace ............................................. 8-29
8.17 XML Elements: AW Cube .............................................................. 8-30
8.18 XML Elements: AW Dimension ..................................................... 8-32
8.19 XML Elements: AW Hierarchy ...................................................... 8-35
8.20 XML Elements: AW Level ............................................................ 8-36
8.21 XML Elements: AW Column ........................................................ 8-37

9 XML Presentation Elements

9.1 XML Elements: Presentation Catalog ............................................ 9-1
9.2 XML Elements: Presentation Schema ............................................. 9-3
9.3 XML Elements: Presentation Table ................................................ 9-3
9.4 XML Elements: Presentation Column ............................................ 9-5
9.5 XML Elements: Presentation Hierarchy .......................................... 9-6
9.6 XML Elements: Presentation Level ................................................ 9-8

10 XML Marketing, Security, and Other Elements

10.1 XML Marketing Elements ............................................................. 10-1
10.1.1 XML Elements: Catalog Link .................................................... 10-1
10.1.2 XML Elements: Target Level .................................................... 10-2
10.1.3 XML Elements: List Catalog .................................................... 10-5
10.1.4 XML Elements: Qualified List Item .......................................... 10-6
10.1.5 XML Elements: Qualifying Key ............................................... 10-8
10.1.6 XML Elements: Sampling Table ............................................... 10-8
10.1.7 XML Elements: Seg Catalog ..................................................... 10-9
10.2 XML Security Elements ............................................................... 10-10
10.2.1 XML Elements: Variable ........................................................ 10-10
10.2.2 XML Elements: LDAP Server ................................................. 10-11
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2.3</td>
<td>XML Elements: Init Block</td>
<td>10-13</td>
</tr>
<tr>
<td>10.2.4</td>
<td>XML Elements: Custom Authenticator</td>
<td>10-16</td>
</tr>
<tr>
<td>10.2.5</td>
<td>XML Elements: Privilege Package</td>
<td>10-17</td>
</tr>
<tr>
<td>10.2.6</td>
<td>XML Elements: Object Privilege</td>
<td>10-19</td>
</tr>
<tr>
<td>10.2.7</td>
<td>XML Elements: Query Privilege</td>
<td>10-20</td>
</tr>
<tr>
<td>10.2.8</td>
<td>XML Elements: User</td>
<td>10-22</td>
</tr>
<tr>
<td>10.2.9</td>
<td>XML Elements: Group</td>
<td>10-23</td>
</tr>
<tr>
<td>10.2.10</td>
<td>XML Elements: DB Sign-On</td>
<td>10-24</td>
</tr>
<tr>
<td>10.2.11</td>
<td>XML Elements: SetPermission</td>
<td>10-25</td>
</tr>
<tr>
<td>10.2.12</td>
<td>XML Elements: FMW Security</td>
<td>10-27</td>
</tr>
<tr>
<td>10.3</td>
<td>Other XML Elements</td>
<td>10-27</td>
</tr>
<tr>
<td>10.3.1</td>
<td>XML Elements: Project</td>
<td>10-27</td>
</tr>
<tr>
<td>10.3.2</td>
<td>XML Elements: Query Object</td>
<td>10-30</td>
</tr>
</tbody>
</table>

**Glossary**

**Index**
Preface

The Oracle Business Intelligence Foundation Suite is a complete, open, and integrated solution for all enterprise business intelligence needs, including reporting, ad hoc queries, OLAP, dashboards, scorecards, and what-if analysis. The Oracle Business Intelligence Foundation Suite includes Oracle Business Intelligence Enterprise Edition.

Oracle Business Intelligence Enterprise Edition (Oracle BI EE) is a comprehensive set of enterprise business intelligence tools and infrastructure, including a scalable and efficient query and analysis server, an ad-hoc query and analysis tool, interactive dashboards, proactive intelligence and alerts, and an enterprise reporting engine.

The components of Oracle BI EE share a common service-oriented architecture, data access services, analytic and calculation infrastructure, metadata management services, semantic business model, security model and user preferences, and administration tools. Oracle BI EE provides scalability and performance with data-source specific optimized request generation, optimized data access, advanced calculation, intelligent caching services, and clustering.

This guide contains usage and reference information about both the Oracle BI Server MDS XML API and the legacy Oracle BI Server XML API.

Audience

This document is intended for metadata repository developers who use MDS XML or Oracle BI Server XML in their repository development processes, or for system administrators who use the Oracle BI Server XML API to programmatically interact with the Oracle BI repository.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
Related Documents and Other Resources

See the Oracle Business Intelligence documentation library for a list of related Oracle Business Intelligence documents.

In addition:

■ Go to the Oracle Learning Library for Oracle Business Intelligence-related online training resources.

■ Go to the Product Information Center support note (Article ID 1267009.1) on My Oracle Support at https://support.oracle.com.

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Part I provides usage and reference information for the Oracle BI Server MDS XML API.

Part I contains the following chapters:

- Chapter 1, "About the Oracle BI Server MDS XML API"
- Chapter 2, "MDS XML Logical Elements"
- Chapter 3, "MDS XML Physical Elements"
- Chapter 4, "MDS XML Presentation Elements"
- Chapter 5, "MDS XML Marketing, Security, and Other Elements"
About the Oracle BI Server MDS XML API

This chapter provides information about Oracle BI Server MDS XML, an XML format that is compatible with Oracle Metadata Services and the Oracle BI repository. Oracle Business Intelligence provides the ability to save Oracle BI repository metadata as a set of XML documents in MDS XML format.


**Note:** MDS XML format is different from the XML format generated by the Oracle BI Server XML API, where the metadata for an Oracle BI repository is generated in a single XML file. For more information about the Oracle BI Server XML API, see Chapter 6, "About the Oracle BI Server XML API."

This chapter contains the following topics:

- Section 1.1, “About the Oracle BI Server MDS XML API”
- Section 1.2, "Generating MDS XML"
- Section 1.3, "Executing MDS XML"
- Section 1.4, "Generic MDS XML Elements and Attributes"
- Section 1.5, "Example: Analysis of an MDS XML Logical Table Element"
- Section 1.6, "Topics of Interest in Other Guides"
- Section 1.7, "System Requirements and Certification"

### 1.1 About the Oracle BI Server MDS XML API

The contents and structure of the Oracle BI Server MDS XML repository files are defined through the following schema element files:

- xudml_mds_admin.xsd and xudml_mds.xsd
  
  These files contain the basic MDS XML grammar for the repository elements.

- xudml_mds_core.xsd
  
  This file contains the detailed MDS XML grammar for the repository elements and attributes and the relationships between the elements.
You can find these files at:

`ORACLE_HOME/bifoundation/server/bin`

You can view these files to see the relationships between MDS XML metadata objects, but you must not modify these files.

### 1.2 Generating MDS XML

You can generate MDS XML from an existing repository file either from within the Administration Tool or by using a command-line utility.

This section contains the following topics:

- Section 1.2.1, "Generating MDS XML from an Existing RPD Using the Administration Tool"
- Section 1.2.2, "Generating MDS XML from an Existing RPD Using a Command-Line Utility"

#### 1.2.1 Generating MDS XML from an Existing RPD Using the Administration Tool

Perform these steps to generate MDS XML from an existing repository file using the Administration Tool:

1. Open your existing repository file (RPD) in the Administration Tool in offline mode.
2. Select File, then select Save As, then select MDS XML Documents.
3. Select a root location for your MDS XML repository files, and then click OK.

The output files are generated in directories under the following path:

```
root_location/oracle/bi/server/base.
```

The number of generated directories depends on the complexity of your repository. For example, with `C:\SampleApp_MDS` selected as your root location, the output directory structure could be as shown in Figure 1–1.
1.2.2 Generating MDS XML from an Existing RPD Using a Command-Line Utility

Oracle Business Intelligence provides the `biserverxmlgen` command-line utility to generate MDS XML output from a repository file.

Before running biserverxmlgen, you must first run bi-init to launch a command prompt or shell window that is initialized to your Oracle instance. This utility is called `bi-init.sh` on Linux and UNIX systems, `bi-init.bat` on client installations of the Administration Tool, and `bi-init.cmd` for all other Windows installation types.

For example, on Windows:

1. In Windows Explorer, go to the location appropriate for your install type:
   - Client installations:
     ```
     ORACLE_HOME/bifoundation/server/bin
     ```
   - All other installations:
     ```
     ORACLE_INSTANCE/bifoundation/OracleBIApplication/coreapplication/setup
     ```

2. Double-click `bi-init.cmd` (or `bi-init.bat` for client installations) to display a command prompt that is initialized to your Oracle instance.

3. At the command prompt, type `biserverxmlgen` with the desired options. You can also run `biserverxmlgen` with no arguments or parameters to see usage information.
To use biserverxmlgen to generate MDS XML output from an existing repository file, execute the following at the command prompt:

```
```

Where:

- `-M` indicates that you want MDS XML output.

`repository_pathname` is the name and location of the repository file from which you want to generate MDS XML.

`password` is the repository password.

The password argument is optional. If you do not provide a password argument, you are prompted to enter a password when you run the command. To minimize the risk of security breaches, Oracle recommends that you do not provide a password argument either on the command line or in scripts. Note that the password argument is supported for backward compatibility only, and will be removed in a future release. For scripting purposes, you can pass the password through standard input.

`output_script_pathname` is the name and location of the XML file created by biserverxmlgen. By default, this file contains XML for all objects. However, if you use the `-C` option, then this file contains only XUDML for connection pools and their relative variables.

`directory_name` is the name and location of the directory under which you want to generate your MDS XML output.

Specify `-8` to use UTF-8 encoding in the generated XML file. If you do not specify `-8`, the generated XML file uses ANSI ISO-8859-1 encoding by default.

Specify `-N` to generate output without upgrade ids (uids). These are the internal identifiers of objects. Use this if you want to create a new repository without the old upgrade ids.

You can use one of the following options to control the generation of security objects, presentation objects, or connection pools. Note if you specify more than one option, then you will receive an error.

- Use `-Q` to generate output without security objects
- Use `-S` to generate output for only security objects
- Use `-L` to generate only a list of presentation objects.
- Use `-C` to generate a script only for connection pools and variables relative to the connection pool.

`variable_list_pathname` is optionally used with `-C`. It is the file name that contains the variable list used to generate the specified variables. If you specify this option, then the following are generated: connection pools, variables relative to the connection pool, and variables from the file.

1.3 Executing MDS XML

You can use the biserverxmlmlexec command-line utility to execute MDS XML in offline mode to create or modify a repository file. This utility can generate an entire repository file (RPD) from a set of MDS XML documents, or it can modify an existing repository file by applying the changes specified in the MDS XML. The input MDS XML is validated against the MDS XML schema. Any objects that exist in both the RPD and
the MDS XML documents are overwritten by the corresponding object definitions in the MDS XML. Note that the same object cannot be defined twice in the MDS XML documents.

Before running biserverxmlxexec, you must first run bi-init to launch a command prompt or shell window that is initialized to your Oracle instance. This utility is called bi-init.sh on Linux and UNIX systems, bi-init.bat on client installations of the Administration Tool, and bi-init.cmd for all other Windows installation types.

For example, on Windows:
1. In Windows Explorer, go to the location appropriate for your install type:
   - Client installations:
     
     \[\text{ORACLE\_HOME/bifoundation/server/bin}\]
   - All other installations:
     
     \[\text{ORACLE\_INSTANCE/bifoundation/OracleBIApplication/coreapplication/setup}\]

2. Double-click bi-init.cmd (or bi-init.bat for client installations) to display a command prompt that is initialized to your Oracle instance.

3. At the command prompt, type biserverxmlxexec with the desired options. You can also run biserverxmlxexec with no arguments or parameters to see usage information.

To use biserverxmlxexec to execute MDS XML, execute the following at the command prompt:

\[\text{biserverxmlxexec -}D \text{MDS\_XML\_document\_directory}\ [-S \text{MDS\_XML\_password}]\]
\[-B \text{base\_repository\_pathname}\] -O \text{output\_repository\_pathname} [-P rpd\_password] [-M] [-E] [-C]\n
Where:

- **MDS\_XML\_document\_directory** is the location of the input MDS XML documents.
- **MDS\_XML\_password** is the password for the set of MDS XML documents. You only need to specify -S when the set of MDS XML documents has a different password than the password specified for -P. Note also that you can specify -C to avoid password checking for the MDS XML documents.

The MDS\_XML\_password argument is optional. If you do not provide an MDS XML password argument, and if the set of MDS XML documents has a different password than the password specified for -P, you are prompted to enter an MDS XML password when you run the command (unless you have specified -C). To minimize the risk of security breaches, Oracle recommends that you do not provide a password argument either on the command line or in scripts. Note that the password argument is supported for backward compatibility only, and will be removed in a future release. For scripting purposes, you can pass the password through standard input.

- **base\_repository\_pathname** is the existing repository file you want to modify using the set of MDS XML documents (optional). Do not specify this argument if you want to generate a new repository file from the MDS XML.

- **rpd\_password** is the password for the RPD file. If you specified a base repository, enter the repository password for the base repository. If you did not specify a base repository, enter the password you want to use for the new repository.

The password argument is optional. If you do not provide a password argument, you are prompted to enter a password when you run the command. To minimize the risk
of security breaches, Oracle recommends that you do not provide a password argument either on the command line or in scripts. Note that the password argument is supported for backward compatibility only, and will be removed in a future release. For scripting purposes, you can pass the password through standard input.

`output_repository_pathname` is the name and location of the RPD output file you want to generate. Note that you must specify an output repository regardless of whether you are generating a new RPD, or whether you are modifying an existing base repository. The changes that result from executing MDS XML on a base repository are contained in the specified output repository, not in the base repository.

Specify `-M` to specify that you want to execute MDS XML documents. If you specify `-D`, the `-M` argument is not needed. You only need to specify `-M` when you have a single MDS XML file that contains all the object definitions.

Specify `-E` to exit without creating the RPD output file if an expression parsing error occurs.

Specify `-C` to skip prompting for the input MDS XML password. When you use this option, all connection pool passwords are removed from the output RPD.

**Example**
The following example creates a new repository called `rp1` based on the input MDS XML document directory:

```
biserverxmlexec -D C:\MDS_dir -O rp1.rpd
Give password: my_rpd_password
```

### 1.4 Generic MDS XML Elements and Attributes

This section describes generic elements and attributes that are common, like `Description`, `name`, and `mdsid`. In the Oracle BI Server MDS XML schema (.xsd file), these elements and attributes are defined under the `object_t` element.

The only common element is the `Description` element. This element can hold a text string describing a particular object. For example:

```
<Description>
  <![CDATA[ Generated by Time Series Wizard ]]> 
</Description>
```

There are nine common attributes defined under the `object_t` element. Table 1–1 lists these attributes.

**Table 1–1  Common MDS XML Attributes in object_t**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>id</td>
<td>Not used for MDS XML.</td>
</tr>
<tr>
<td>mdsid</td>
<td>mdsid is a universally unique identifier, 37-characters long.</td>
</tr>
<tr>
<td>iconIndex</td>
<td>The index number for the icon of the object in the Administration Tool.</td>
</tr>
<tr>
<td>isDefaultReadable</td>
<td>If set to <code>true</code>, indicates that this object is readable by default.</td>
</tr>
<tr>
<td>isDefaultReadWriteable</td>
<td>If set to <code>true</code>, indicates that this object is both readable and writable by default.</td>
</tr>
<tr>
<td>isDefaultDeletable</td>
<td>If set to <code>true</code>, indicates that this object is deletable by default.</td>
</tr>
</tbody>
</table>
In addition to the common attributes defined in object_t, an additional common attribute is defined in ref_t. This attribute is used when one object references another object or set of objects. For example, business model objects reference the logical display folders contained in that business model, using the sub-element RefLogicalDisplayFolder.

All referenced sub-elements have the same common attribute refId shown in Table 1–2.

### Table 1–2 Common MDS XML Attribute in ref_t

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refId</td>
<td>Consists of the mdsid of the referencing object and the mdsid of the referenced object, separated by &quot;.&quot;.</td>
</tr>
</tbody>
</table>

#### 1.4.1 About MDS XML Object Identifiers

Oracle BI Server MDS XML metadata objects have a single unique identifier:

- **mdsid**: This identifier consists of 37 characters, and is unique across the entire repository. The mdsid is created by the Oracle BI Server for the lifetime of the object.

  For example: mdsid="m1d08ed10-0bf4-1000-806e-0a9716a80000".

#### 1.5 Example: Analysis of an MDS XML Logical Table Element

This section illustrates the use of MDS XML identifiers and reference attributes for a Logical Table called F2 Coupon Usages.

The logical table F2 Coupon Usages has the following properties:

- It contains four logical columns, including the column ORDER_ID
- It has a logical key CUSTOMER_ID, that consists of the single logical column ORDER_ID
- It is contained within the business model Fusion Order Demo
- It has one logical table source COUPON_USAGES

The MDS XML elements to illustrate this example occur in three separate MDS XML files: a file that describes the logical table elements and two files that describe the associated business model and logical table source elements.

The focus of this example is the logical table file, especially the references in the file to related elements and attributes. The only details shown of the business model and logical table source files are those that illustrate relationships with the logical table.

As an overview, the following figure shows these three files highlighted within the general directory structure created when MDS XML is generated from an Oracle BI repository.

---

**Table 1–1 (Cont.) Common MDS XML Attributes in object_t**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDefaultGrantable</td>
<td>If set to true, indicates that this object is grantable by default</td>
</tr>
<tr>
<td>isDefaultAll</td>
<td>If set to true, indicates that this object is readable, writable, deletable, and grantable by default.</td>
</tr>
</tbody>
</table>
The contents of the main Logical Table file
F2 Coupon Usages_80cbbde8-07f3-0000-714b-e31d00000000.xml
appear in this section, with the mdsid of the logical table highlighted (the italicized line numbers have been inserted for easy element tag identification in the subsequent sections of this topic).

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<LogicalTable mdsid="m80cbbde8-07f3-0000-714b-e31d00000000"
   name="F2 Coupon Usages"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.oracle.com/obis/repository"
   x="133" y="378"
   subjectAreaRef="/oracle/bi/server/base/BusinessModel
   /Fusion Order Demo_80cbbdd2-07d0-0000-714b-e31d00000000.xml#"
   m80cbbdd2-07d0-0000-714b-e31d00000000">
  <LogicalColumn mdsid="m80cbbeff-07d6-0000-714b-e31d00000000"
     name="Applied Date" isWriteable="false">
  </LogicalColumn>
  <LogicalColumn mdsid="m80cbbf00-07d6-0000-714b-e31d00000000"
     name="# of Customers" isWriteable="false">
  </LogicalColumn>
  <LogicalColumn mdsid="m80cbbf04-07d6-0000-714b-e31d00000000"
     name="Order ID" isWriteable="false">
  </LogicalColumn>
  <LogicalKey mdsid="m80cbbf39-07d8-0000-714b-e31d00000000"
     name="CUSTOMER_ID" isPrimary="true">
    <RefColumns/>
    <RefLogicalColumn refId="m80cbbf39-07d8-0000-714b-e31d00000000-
     m80cbbf04-07d6-0000-714b-e31d00000000"
     logicalColumnRef="/oracle/bi/server/base/LogicalTable
     /F2 Coupon Usages_80cbbde8-07f3-0000-714b-e31d00000000.xml#"
     m80cbbf04-07d6-0000-714b-e31d00000000" />
</LogicalTable>
```
Example: Analysis of an MDS XML Logical Table Element

Reference from Logical Table to Business Model

The subjectAreaRef attribute (lines 7 to 9) points to the file and mdsid that describe the related business model:

```
subjectAreaRef="/oracle/bi/server/base/BusinessModel
/Fusion Order Demo_80cbbdd2-07d0-0000-714b-e31d00000000.xml#
```

Note that the file name includes most of the business model mdsid.

The Business Model file (full contents not shown here) contains the BusinessModel tag, which includes the Business Model mdsid attribute:

```
<BusinessModel mdsid="m80cbbdd2-07d0-0000-714b-e31d00000000"
name="Fusion Order Demo" .....`

Reference from Logical Key to Logical Column (in current Logical Table)

The logical key tag (lines 22 to 32) contains the mdsid of the logical key.

```
<LogicalKey mdsid="m80cbbf39-07d8-0000-714b-e31d00000000"
name="CUSTOMER_ID" ...
```

In general, a logical key can consist of several logical columns. In this case, the logical key, CUSTOMER_ID, consists of one logical column, Order ID.

The logical column of the logical key CUSTOMER_ID is referenced in the RefColumns tag (lines 24 to 31), and specifically the RefLogicalColumn sub-element.

```
<RefColumns>
RefLogicalColumn
refId="m80cbbf39-07d8-0000-714b-e31d00000000-
m80cbbf04-07d6-0000-714b-e31d00000000"
logicalColumnRef="/oracle/bi/server/base/LogicalTable
/F2 Coupon Usages_80cbbd8e-07f3-0000-714b-e31d00000000.xml#
```

logicalColumnRef (lines 28 to 30) points to the current Logical Table file (lines 28 to 29) and the mdsid that describe the logical column within the logical table that makes up the logical key (line 30).

The logical column mdsid (line 30) references the mdsid of the LogicalColumn tag (lines 19 to 21), that appears just before the LogicalKey tag in the current Logical Table file:

```
<LogicalColumn mdsid="m80cbbf04-07d6-0000-714b-e31d00000000"
name="Order ID" ...
```
The refId value of the RefLogicalColumn sub-element of the logical table (lines 26 to 27):

```
26  refId="m80cbbf39-07d8-0000-714b-e31d00000000-
27   m80cbbf04-07d6-0000-714b-e31d00000000"
```

consists of the mdsid of the logical key itself (line 22 of the logical table file, shown in bold at the beginning of this topic) and the mdsid of the logical column to which it refers (line 19 of the logical table file, shown in bold in the main body of this topic), separated by "-".

**Reference from Logical Table to Logical Table Source**

In general, a logical table can refer to several logical table sources. In this case, the logical table F2 Coupon Usages refers to the single logical table source COUPON_USAGES.

The RefTableSources tag (lines 33 to 41) is a child element of the LogicalTable tag. The explicit reference to the logical table source appears within the RefLogicalTableSource sub-element.

```
33  <RefTableSources>
34    <RefLogicalTableSource
35      refId="m80cbbde8-07f3-0000-714b-e31d00000000-
36        m80cbc055-07f5-0000-714b-e31d00000000"
37      logicalTableSourceRef=
38        "oracle/bi/server/base/LogicalTableSource
39         /COUPON_USAGES_80cbc055-07f5-0000-714b-e31d00000000.xml#
40        m80cbc055-07f5-0000-714b-e31d00000000" />
41  </RefTableSources>
```

logicalTableSourceRef (lines 37 to 40) points to the file and the mdsid that describe the logical table source (note that the file name includes most of the logical table source mdsid).

The Logical Table Source file (full contents not shown here) contains the LogicalTableSource tag, which includes the mdsid of the logical table source:

```
<LogicalTableSource mdsid="m80cbc055-07f5-0000-714b-e31d00000000"
   name="COUPON_USAGES" ..."
```

The refId value of the RefLogicalTableSource sub-element of the logical table (lines 35 to 36):

```
35  refId="m80cbbde8-07f3-0000-714b-e31d00000000-
36    m80cbc055-07f5-0000-714b-e31d00000000"
```

consists of the mdsid of the logical table itself (shown in bold in line 2 of the logical table file listing, at the beginning of Section 1.5, "Example: Analysis of an MDS XML Logical Table Element") and the mdsid of the logical table source (shown in bold in the body of this topic), separated by "-".

### 1.6 Topics of Interest in Other Guides

Some topics that may be of interest to XML schema reference users are covered in other guides. Table 1–3 lists these topics, and indicates where to go for more information.
1.7 System Requirements and Certification

Refer to the system requirements and certification documentation for information about hardware and software requirements, platforms, databases, and other information. Both of these documents are available on Oracle Technology Network (OTN).

The system requirements document covers information such as hardware and software requirements, minimum disk space and memory requirements, and required system libraries, packages, or patches:


The certification document covers supported installation types, platforms, operating systems, databases, JDKs, and third-party products:

This chapter provides MDS XML reference information for logical elements. Logical elements typically correspond to objects in the Business Model and Mapping layer of the Oracle BI repository, or in other words, the objects that appear in the center pane of the Oracle BI Administration Tool.

This chapter contains the following topics:
- Section 2.1, "MDS XML Elements: Logical Column"
- Section 2.2, "MDS XML Elements: Business Model"
- Section 2.3, "MDS XML Elements: Logical Complex Join"
- Section 2.4, "MDS XML Elements: Logical Level"
- Section 2.5, "MDS XML Elements: Dimension"
- Section 2.6, "MDS XML Elements: Logical Table"
- Section 2.7, "MDS XML Elements: Logical Foreign Key"
- Section 2.8, "MDS XML Elements: Logical Key"
- Section 2.9, "MDS XML Elements: Logical Table Source"
- Section 2.10, "MDS XML Elements: Logical Source Folder"
- Section 2.11, "MDS XML Elements: Measure Definition"
- Section 2.12, "MDS XML Elements: Aggregation Rule"
- Section 2.13, "MDS XML Elements: Override Aggregation Rule"
- Section 2.14, "MDS XML Elements: Logical Display Folder"

2.1 MDS XML Elements: Logical Column

The logical column element corresponds to the logical column object in the Business Model and Mapping layer.

The logical column element is not a top-level element - it appears as a child element of the logical table element.

The logical column tag includes the following attributes:
**Table 2–1 Logical Column Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDerived</td>
<td>When set to true, indicates that the logical column is based on a user-defined expression (for example, another logical column, a constant value, or a variable).</td>
</tr>
<tr>
<td>IsWriteable</td>
<td>When set to true, indicates that write back has been enabled for this column. This feature is typically used with ADF Business Component data sources. This option is called <strong>Writeable</strong> in the Administration Tool.</td>
</tr>
<tr>
<td>customSortOrderRef</td>
<td>If the sort order for this logical column is based on a different logical column, this attribute references that logical column.</td>
</tr>
<tr>
<td>descriptorIDRef</td>
<td>When multilingual columns are based on a lookup function, it is common to specify the non-translated lookup key column as the descriptor ID column of the translated column. This attribute references the descriptor ID column.</td>
</tr>
</tbody>
</table>

The logical column tag includes the following child elements:

**Table 2–2 Logical Column Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expr</td>
<td>Contains the expression from which the logical column should be derived. This element is only used when the <strong>isDerived</strong> attribute is set to true.</td>
</tr>
<tr>
<td>MeasureDefn</td>
<td>Contains the measure definition for a measure column. See Section 2.11, &quot;MDS XML Elements: Measure Definition&quot; for more information.</td>
</tr>
<tr>
<td>AttrDefn</td>
<td>Specifies the internal object to cache the logical column's mapping to the physical layer.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<LogicalColumn ...
    isDerived="true"
    IsWriteable="false"
    customSortOrderRef="" 
    descriptorIDRef="">
    <Description>
        <![CDATA[ "Days Closed Category by number of days closed. Categories are, A: <= 5, B: 6 - 30, and C: 31+" ]]>
    </Description>
    <Expr>
        <![CDATA[ CASE WHEN "%1" IS NULL THEN 'A: <=5' WHEN "%1" <= 5 THEN 'A: <= 5' ]]>
    </Expr>
</LogicalColumn>
```

**Example**

```xml
<LogicalColumn mdsid="m8030eac4-07d6-0000-714b-e31d00000000"
    name="Days Closed Category"
    isDerived="true"
    isWriteable="false">
    <Description>
        <![CDATA[ "Days Closed Category by number of days closed. Categories are, A: <= 5, B: 6 - 30, and C: 31+" ]]>
    </Description>
    <Expr>
        <![CDATA[ CASE WHEN "%1" IS NULL THEN 'A: <=5' WHEN "%1" <= 5 THEN 'A: <= 5' ]]>
    </Expr>
</LogicalColumn>
```
WHEN "%1" <= 30 THEN 'B: 6 - 30'
ELSE 'C: 31+' END ">
</ExprText>
<ExprTextDesc>
<![CDATA[
WHEN "Contact List"."Service Request"."Days Closed" IS NULL THEN 'A: <=5'
WHEN "Contact List"."Service Request"."Days Closed" <= 5 THEN 'A: <= 5'
WHEN "Contact List"."Service Request"."Days Closed" <= 30 THEN 'B: 6 - 30'
ELSE 'C: 31+' END 
]]>
</ExprTextDesc>
<ObjectRefList>
<RefObject refId="m8030eac4-07d6-0000-714b-e31d00000000-m8030eac2-07d6-0000-714b-e31d00000000" objectTypeId="2006" objectRef="/oracle/bi/server/base/LogicalTable/Service Request_8030ea6a-07f3-0000-714b-e31d00000000.xml#m8030eac2-07d6-0000-714b-e31d00000000"/>
</ObjectRefList>
</Expr>
</LogicalColumn>

2.2 MDS XML Elements: Business Model

The business model element corresponds to the business model object in the Business Model and Mapping layer.

The business model tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isClassicStar</td>
<td>If set to true, indicates that the business model follows a star schema. Star schemas have one-to-many relationships between the logical dimension tables and the logical fact table.</td>
</tr>
<tr>
<td>isAvailable</td>
<td>If set to true, indicates that the corresponding business model is available for queries.</td>
</tr>
</tbody>
</table>

The business model tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefDisplayFolders</td>
<td>References the display folders that belong to this business model. Each display folder is contained in a sub-element called RefLogicalDisplayFolder.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<BusinessModel ...
    isClassicStar="" 
    isAvailable=""/>
<Description />
<RefDisplayFolders>
    <RefLogicalDisplayFolder logicalDisplayFolderRef="" .../>
    <RefLogicalDisplayFolder logicalDisplayFolderRef="" .../>
</RefDisplayFolders>
</BusinessModel>
```
Example

```xml
<BusinessModel mdsid="m1d08ed10-0bf4-1000-806e-0a9716a80000"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    isClassicStar="false"
    isAvailable="true">
  <RefDisplayFolders>
    <RefLogicalDisplayFolder refId="m1d08ed10-0bf4-1000-806e-0a9716a80000-
        m1d0ae8e0-0bf4-1000-806e-0a9716a80000"
        logicalDisplayFolderRef="/oracle/bi/server/base/LogicalDisplayFolder/F1_
        ld0ae8e0-0bf4-1000-806e-0a9716a80000.xml#m1d0ae8e0-0bf4-1000-806e-
        0a9716a80000" />
    <RefLogicalDisplayFolder refId="m1d08ed10-0bf4-1000-806e-0a9716a80000-
        m4edbf441-0bf9-1000-ala7-0a970e580000"
        logicalDisplayFolderRef="/oracle/bi/server/base/LogicalDisplayFolder/F2_
        4edbf441-0bf9-1000-ala7-0a970e580000.xml#m4edbf441-0bf9-1000-ala7-
        0a970e580000" />
  </RefDisplayFolders>
</BusinessModel>
```

2.3 MDS XML Elements: Logical Complex Join

The logical complex join element corresponds to the logical join object in the Business Model and Mapping Layer.

The logical complex join tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of join. Valid values are Inner, LeftOuter, RightOuter, FullOuter, and FullOuterStitch.</td>
</tr>
<tr>
<td>drivingTableRef</td>
<td>If a driving table has been specified for the key, this attribute references that logical table.</td>
</tr>
<tr>
<td>logicalTable1Ref</td>
<td>References the first logical table of the join.</td>
</tr>
<tr>
<td>multiplicity1</td>
<td>multiplicity1 and multiplicity2 indicate the multiplicity of the tables in the join. Valid values are ((0-9)^<em>) TO ((0-9)^</em>), ZERO TO ONE, ZERO TO MANY, ONE TO ONE, ONE TO MANY, and UNKNOWN.</td>
</tr>
<tr>
<td>isAggregate1</td>
<td>isAggregate1 and isAggregate2 indicate whether the corresponding table is an aggregate.</td>
</tr>
<tr>
<td>logicalTable2Ref</td>
<td>References the second logical table of the join.</td>
</tr>
<tr>
<td>multiplicity2</td>
<td>(See Description for multiplicity1.)</td>
</tr>
<tr>
<td>isAggregate2</td>
<td>(See Description for isAggregate1.)</td>
</tr>
</tbody>
</table>

Syntax

```xml
<LogicalComplexJoin _
    type=""
    drivingTableRef=""
    logicalTable1Ref=""
    multiplicity1=""
    isAggregate1=""
    logicalTable2Ref=""
    multiplicity2=""```
Example

```xml
<LogicalComplexJoin mdsid="m80e70525-07d4-0000-714b-e31d00000000"
    name="Relationship_2004:9974601258446906"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    type="Inner"
    logicalTable1Ref="/oracle/bi/server/base/LogicalTable/Dim Rating Model 80d63a05-07f3-0000-714b-e31d00000000.xml#m80d63a05-07f3-0000-714b-e31d00000000"
    multiplicity1="0..1"
    isAggregate1="false"
    logicalTable2Ref="/oracle/bi/server/base/LogicalTable/Fact Performance Rating_80d12881-07f3-0000-714b-e31d00000000.xml#m80d1288107f3-0000-714b-e31d00000000"
    multiplicity2="0..n"
    isAggregate2="false">
    <Description />
</LogicalComplexJoin>
```

2.4 MDS XML Elements: Logical Level

The logical level element corresponds to the logical level object contained in the dimensions in the Business Model and Mapping layer.

The logical level element is not a top-level element - it appears as a component element of the dimension element.

The logical level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>isUniqueSourceKey</code></td>
<td>If set to true, indicates that this level is associated with the logical column or columns that comprise the logical key of the dimension table (or in other words, that this level is the lowest level of the dimension).</td>
</tr>
</tbody>
</table>
| `isGTA`           | If set to true, indicates that this level is the Grand Total level, which is a special level representing the grand total for a dimension. Each dimension can have just one Grand Total level. A grand total level does not contain dimensional attributes and does not have a level key.  
Note: GTA stands for Grand Total Alias. |
| `isFDUC`          | If set to true, indicates that the logical level rolls up to its parent. This option is called Supports rollup to higher level of aggregation in the Administration Tool.  
Note: FDUC stands for FullDrillUpCoverage. |
| `levelConst`      | The star schema level constant.                                             |
| `memberCount`     | The number of elements that exist at this logical level.                    |

The logical level tag includes the following child elements:
**Table 2–7 Logical Level Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Logical Key           | References the parent key for this level (for dimensions with parent-child hierarchies only).  
                        | The parent key is contained in a sub-element called RefColumns. For details, see Section 2.8, "MDS XML Elements: Logical Key." |
| RefChildLevels        | References the child levels that have been defined for this logical level.  
                        | Each child level is contained in a sub-element called RefLogicalLevel.                                                |
| RefPreferredDrillDown | If a drill path has been defined that is outside the normal drill path defined by the dimension level hierarchy, this element references the level to which users should drill.  
                        | The preferred drill level is contained in a sub-element called RefLogicalLevel.                                                |
| RefLogicalColumns     | References the logical columns associated with this logical level.  
                        | Each logical column is contained in a sub-element called RefLogicalColumn.                                                |

**Syntax**

The syntax here includes the summary child element entry for the logical key element. The full syntax for the logical key element appears in Section 2.8, "MDS XML Elements: Logical Key."

```xml
<LogicalLevel ...
   isUniqueSourceKey="" 
   isGTA=""
   isFDUC=""
   levelConst=""
   memberCount="">

   <Description />
   <LogicalKey />
   <RefChildLevels>
      <RefLogicalLevel logicalLevelRef="" ...
   </RefChildLevels>
  </RefChildLevels>

  <RefPreferredDrillDown>
    <RefLogicalLevel logicalLevelRef="" ...
  </RefPreferredDrillDown>

  <RefLogicalColumns>
    <RefLogicalColumn logicalColumnRef="" ...
  </RefLogicalColumns>

</LogicalLevel>
```

**Example**

```xml
<LogicalLevel mdsid="m80334c74-07e9-0000-714b-e31d00000000"
   name="Year"
   isGTA="false"
   isFDUC="true"
   memberCount="1"
   levelConst="Year">

   <LogicalKey mdsid="m80334d04-07d8-0000-714b-e31d00000000"
      name="Last Login Year"
      isPrimary="true"
      isForDrillDown="true">

   <RefColumns>
```

---

*Table 2–7 Logical Level Child Elements* (continued)

**Element Name**

**Description**

| Logical Key | References the parent key for this level (for dimensions with parent-child hierarchies only).  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The parent key is contained in a sub-element called RefColumns. For details, see Section 2.8, &quot;MDS XML Elements: Logical Key.&quot;</td>
</tr>
</tbody>
</table>
| RefChildLevels | References the child levels that have been defined for this logical level.  
|               | Each child level is contained in a sub-element called RefLogicalLevel.                                                |
| RefPreferredDrillDown | If a drill path has been defined that is outside the normal drill path defined by the dimension level hierarchy, this element references the level to which users should drill.  
|                       | The preferred drill level is contained in a sub-element called RefLogicalLevel.                                                |
| RefLogicalColumns | References the logical columns associated with this logical level.  
|                   | Each logical column is contained in a sub-element called RefLogicalColumn.                                                |
2.5 MDS XML Elements: Dimension

The dimension element corresponds to the dimension object in the Business Model and Mapping layer.

The dimension tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isTimeDim</td>
<td>If set to true, indicates that this dimension is a time dimension.</td>
</tr>
<tr>
<td>isRagged</td>
<td>If set to true, indicates that this dimension hierarchy is unbalanced. An unbalanced hierarchy is one that contains levels that are not at the same depth.</td>
</tr>
<tr>
<td>isSkipped</td>
<td>If set to true, indicates that this dimension hierarchy is a skip-level hierarchy. A skip-level hierarchy is one in which there are members that do not have a value for a particular parent level.</td>
</tr>
<tr>
<td>isValueBased</td>
<td>If set to true, indicates that this object is a dimension with parent-child hierarchies. A parent-child hierarchy is a hierarchy of members that all have the same type. In a parent-child hierarchy, the inter-member relationships are parent-child relationships between dimension members.</td>
</tr>
<tr>
<td>defaultRootLevelRef</td>
<td>References the default root level of this dimension. When there are many root levels in a dimension, the default root level is the one that is used for drilldown.</td>
</tr>
<tr>
<td>subjectAreaRef</td>
<td>References the business model of this dimension.</td>
</tr>
</tbody>
</table>

The dimension tag includes the following child elements:
### Syntax

The syntax here includes the summary child element entry for the logical level element. The full syntax for the logical level element appears in Section 2.4, "MDS XML Elements: Logical Level."

```xml
<Dimension 
   isTimeDim="" 
   isRagged="" 
   isSkipped="" 
   isValueBased="" 
   defaultRootLevelRef="" 
   subjectAreaRef="">
   <Description /> 
   <LogicalLevel />
</Dimension>
```

### Example

```xml
<Dimension mdsid="m80be897c-07e3-0000-714b-e31d00000000" 
   name="Time" 
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 
   xmlns="http://www.oracle.com/obis/repository" 
   isValueBased="false" 
   isRagged="false" 
   isSkipped="false" 
   defaultRootLevelRef="/oracle/bi/server/base/Dimension/80be897c-07e3-0000-714b-e31d00000000" 
   subjectAreaRef="/oracle/bi/server/base/BusinessModel/80be895f-07e3-0000-714b-e31d00000000">
   <LogicalLevel mdsid="m80be8d55-07e9-0000-714b-e31d00000000" 
   name="Year" isGTA="false" isFDUC="true" 
   memberCount="1" levelConst="&apos;Year&apos;">
   <LogicalKey mdsid="m80be9271-07d8-0000-714b-e31d00000000" 
   name="Year" isPrimary='true' isForDrillDown="true">
   <RefColumns>
   <RefLogicalColumn refId="m80be9271-07d8-0000-714b-e31d00000000-m80be8ba5-07d6-0000-714b-e31d00000000" 
   logicalColumnRef="/oracle/bi/server/base/LogicalTable/80be8ba4-07f3-0000-714b-e31d00000000.xml#m80be8ba5-07d6-0000-714b-e31d00000000"/>
   </RefColumns>
   </LogicalKey>
   </LogicalLevel>
</Dimension>
```

### Table 2–9 Dimension Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogicalLevel</td>
<td>References the logical levels for this dimension. For details, see Section 2.4, &quot;MDS XML Elements: Logical Level.&quot;</td>
</tr>
</tbody>
</table>
2.6 MDS XML Elements: Logical Table

The logical table element corresponds to the logical table object in the Business Model and Mapping layer.

The logical table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isLookupTable</td>
<td>If set to true, indicates that this logical table is a lookup table. Lookup tables are typically used to store fields that contain translations in several languages. Queries that require multilingual data join the base table and lookup table to obtain the translated values for each row in the base table.</td>
</tr>
<tr>
<td>x</td>
<td>The x coordinate of the table when it is shown in the business model diagrams.</td>
</tr>
<tr>
<td>y</td>
<td>The y coordinate of the table when it is shown in the business model diagrams.</td>
</tr>
<tr>
<td>subjectAreaRef</td>
<td>References the business model for this logical table.</td>
</tr>
<tr>
<td>levelColumnRef</td>
<td>References the column that is used in the level that corresponds to this logical table.</td>
</tr>
</tbody>
</table>

The logical table tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogicalColumn</td>
<td>Specifies the logical column that belongs to this logical table. For details, see Section 2.1, &quot;MDS XML Elements: Logical Column.&quot;</td>
</tr>
</tbody>
</table>
MDS XML Elements: Logical Table

Syntax
The syntax here includes summary child element entries for the logical column, logical key, and logical foreign key elements. The full syntax for each child element appears in the appropriate section listed in the following table:

<table>
<thead>
<tr>
<th>Table 2–12 References to Child Elements of Logical Table Element</th>
<th>Section Containing Child Element Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child Element Entry</td>
<td>Section Containing Child Element Syntax</td>
</tr>
<tr>
<td>LogicalColumn</td>
<td>MDS XML Elements: Logical Column</td>
</tr>
<tr>
<td>LogicalKey</td>
<td>MDS XML Elements: Logical Key</td>
</tr>
<tr>
<td>LogicalForeignKey</td>
<td>MDS XML Elements: Logical Foreign Key</td>
</tr>
</tbody>
</table>

Example

```xml
<LogicalTable mdsid="me406bc22-0bf1-1000-b290-0ae593200000"
    name="SalesPlan"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    x="170"
    y="97"
    subjectAreaRef="/oracle/bi/server/base/BusinessModel/SnowflakeSales_e406baec-
```
<LogicalTable mdsid="me406bc22-0bf1-1000-b290-0ae593200000">
  <LogicalColumn mdsid="me406bc23-0bf1-1000-b290-0ae593200000" name="Category ID" isWriteable="false" />
  <LogicalColumn mdsid="me406bc24-0bf1-1000-b290-0ae593200000" name="Country" isWriteable="false" />
  <LogicalColumn mdsid="me406bc25-0bf1-1000-b290-0ae593200000" name="Year" isWriteable="false" />
  <LogicalColumn mdsid="me406bc26-0bf1-1000-b290-0ae593200000" name="PlanRevenue" isWriteable="false" />
  <LogicalKey mdsid="me4070952-0bf1-1000-b290-0ae593200000" name="Category ID, Country, Year" isPrimary="true" isForDrillDown="true">
    <RefColumns>
      <RefLogicalColumn refId="me4070952-0bf1-1000-b290-0ae593200000-me406bc23-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc23-0bf1-1000-b290-0ae593200000" />
      <RefLogicalColumn refId="me4070952-0bf1-1000-b290-0ae593200000-me406bc24-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc24-0bf1-1000-b290-0ae593200000" />
      <RefLogicalColumn refId="me4070952-0bf1-1000-b290-0ae593200000-me406bc25-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc25-0bf1-1000-b290-0ae593200000" />
    </RefColumns>
  </LogicalKey>
  <LogicalForeignKey mdsid="me4070a81-0bf1-1000-b290-0ae593200000" name="SalesPlan_FKey" type="Inner" counterPartKeyRef="/oracle/bi/server/base/LogicalTable/Year_e406bbff-0bf1-1000-b290-0ae593200000.xml#me407094c-0bf1-1000-b290-0ae593200000">
    <RefColumns>
      <RefLogicalColumn refId="me4070a81-0bf1-1000-b290-0ae593200000-me406bc25-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc25-0bf1-1000-b290-0ae593200000" />
    </RefColumns>
  </LogicalForeignKey>
  <LogicalForeignKey mdsid="me4070a82-0bf1-1000-b290-0ae593200000" name="SalesPlan_FKey1" type="Inner" counterPartKeyRef="/oracle/bi/server/base/LogicalTable/Category_e406bbb2-0bf1-1000-b290-0ae593200000.xml#me407093e-0bf1-1000-b290-0ae593200000">
    <RefColumns>
      <RefLogicalColumn refId="me4070a82-0bf1-1000-b290-0ae593200000-me406bc23-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc23-0bf1-1000-b290-0ae593200000" />
    </RefColumns>
  </LogicalForeignKey>
  <LogicalForeignKey mdsid="me4070a83-0bf1-1000-b290-0ae593200000" name="SalesPlan_FKey2" type="Inner" counterPartKeyRef="/oracle/bi/server/base/LogicalTable/EmployeeCountry_e406bbee-0bf1-1000-b290-0ae593200000.xml#me4070949-0bf1-1000-b290-0ae593200000">
    <RefColumns>
      <RefLogicalColumn refId="me4070a83-0bf1-1000-b290-0ae593200000-me406bc24-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc24-0bf1-1000-b290-0ae593200000" />
    </RefColumns>
  </LogicalForeignKey>
</LogicalTable>
2.7 MDS XML Elements: Logical Foreign Key

The logical foreign key element corresponds to the logical foreign key object in the Business Model and Mapping layer.

The logical foreign key element is not a top-level element - it appears as a component element of the logical table element.

The logical foreign key tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of join. Valid values are Inner, LeftOuter, RightOuter, FullOuter, and FullOuterStitch.</td>
</tr>
<tr>
<td>counterPartKeyRef</td>
<td>References the logical key in the primary key table that matches the RefColumns of this foreign key.</td>
</tr>
<tr>
<td>drivingTableRef</td>
<td>If a driving table has been specified for this logical foreign key, this element references that logical table.</td>
</tr>
</tbody>
</table>

The logical foreign key tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefColumns</td>
<td>References the logical columns (in the logical table that contains the foreign key) that match counterPartKeyRef. Each logical column is contained in a sub-element called RefLogicalColumn.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<LogicalForeignKey _
  type=""
  counterPartKeyRef=""
  drivingTableRef="">  
  <Description />  
  <RefColumns>  
    <Ref_logicalColumn logicalColumnRef="" _.>/  
    <Ref_logicalColumn logicalColumnRef="" _.>/  
  </RefColumns>  
</LogicalForeignKey>
```
Example

```xml
LogicalForeignKey mdsid="me4070a83-0bf1-1000-b290-0ae593200000"
  name="SalesPlan_FKey2" type="Inner"
  counterPartKeyRef="/oracle/bi/server/base/LogicalTable/EmployeeCountry_e406bbee-0bf1-1000-b290-0ae593200000.xml#me4070949-0bf1-1000-b290-0ae593200000"/>
  <RefColumns>
    <RefLogicalColumn refId="me4070a83-0bf1-1000-b290-0ae593200000-me406bc24-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc24-0bf1-1000-b290-0ae593200000"/>
  </RefColumns>
</LogicalForeignKey>
```

2.8 MDS XML Elements: Logical Key

The logical key element corresponds to the logical key object in the Business Model and Mapping layer.

The logical key element is not a top-level element - it appears as a child element of the logical level and logical table elements.

The logical key tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isPrimary</td>
<td>If set to true, indicates that this key is the logical primary key for the table.</td>
</tr>
<tr>
<td>isForDrillDown</td>
<td>If set to true, indicates that this key is the key that is displayed when a user clicks to drill down.</td>
</tr>
<tr>
<td>isChronKey</td>
<td>If set to true, indicates that this key is a chronological key for a time dimension.</td>
</tr>
<tr>
<td>logicalLevelRef</td>
<td>References the logical level for this logical key.</td>
</tr>
</tbody>
</table>

The logical key tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefColumns</td>
<td>References the logical columns that define the key of the logical table. Each logical column is contained in a sub-element called RefLogicalColumn.</td>
</tr>
</tbody>
</table>

**Example**

```xml
LogicalForeignKey mdsid="me4070a83-0bf1-1000-b290-0ae593200000"
  name="SalesPlan_FKey2" type="Inner"
  counterPartKeyRef="/oracle/bi/server/base/LogicalTable/EmployeeCountry_e406bbee-0bf1-1000-b290-0ae593200000.xml#me4070949-0bf1-1000-b290-0ae593200000">
  <RefColumns>
    <RefLogicalColumn refId="me4070a83-0bf1-1000-b290-0ae593200000-me406bc24-0bf1-1000-b290-0ae593200000" logicalColumnRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406bc22-0bf1-1000-b290-0ae593200000.xml#me406bc24-0bf1-1000-b290-0ae593200000"/>
  </RefColumns>
</LogicalForeignKey>
```
2.9 MDS XML Elements: Logical Table Source

The logical table source element corresponds to the logical table source object in the Business Model and Mapping layer. See also the column mapping element, described in Section 2.9.1, "Column Mapping."

The logical table source tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDistinct</td>
<td>This attribute is used if the values for the logical table source are unique. In the Administration Tool, this option is called <strong>Select distinct values</strong> in the Content tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>isDimOnly</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>isSysGen</td>
<td>If set to true, indicates that an internal utility such as the Aggregate Persistence Wizard has generated this logical table source.</td>
</tr>
<tr>
<td>canCombine</td>
<td>If set to true, indicates that this logical table source can be combined with other logical table sources to provide a full set of results. In the Administration Tool, this option is called <strong>This source should be combined with other sources at this level</strong> in the Content tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>isActive</td>
<td>If set to true, indicates that this logical table source is active. In the Administration Tool, this attribute corresponds with the <strong>Disabled</strong> option in the General tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>priorityGroup</td>
<td>Indicates the priority group number of this logical table source. Logical table source priority group numbers are assigned to indicate which logical table source should be used for queries for which there is more than one logical table source that can satisfy the requested set of columns.</td>
</tr>
<tr>
<td>allowUnmappedTable</td>
<td>Determines if the logical table source can have physical tables which are not mapped to logical columns. This can occur with a snowflake of physical tables A-&gt;B-&gt;C, where a logical table maps to columns in A and C, but B needs to be included in the logical table source because it is in the join path between A and C.</td>
</tr>
<tr>
<td>isJoinRestricted</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>logicalTableRef</td>
<td>References the logical table for this logical table source.</td>
</tr>
</tbody>
</table>
The logical table source tag includes the following child elements:

**Table 2–17 (Cont.) Logical Table Source Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>closureTableRef</td>
<td>References the parent-child table, if any.</td>
</tr>
</tbody>
</table>

**Table 2–18 Logical Table Source Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColumnMapping</td>
<td>For details, see Section 2.9.1, &quot;Column Mapping.&quot;</td>
</tr>
</tbody>
</table>
| Link               | References the set of physical elements (physical tables, cube tables, AW cubes, or AW dimensions) to which the logical table source is mapped. These links are references to the physical joins that already exist. In the Administration Tool, this element corresponds with the Map to these tables section in the General tab of the Logical Table Source dialog. Each Link child element contains two sub-elements:  
  - The startNodeTableRef attribute references the physical element identified by this link. This can be a Cube Table, Physical Table, AW Dimension, or AW Cube, depending on your data source type.  
  - The Joins sub-element defines the joins for this link. It contains different sub-elements, depending on the type of join. For example:  
    - RefCubeTable, RefPhysicalTable, RefAWCube, or RefAWDimension  
    - RefPhysicalForeignKey or RefComplexJoin  
    - DrivingTable, which contains a sub-element RefPhysicalTable  
  
  The Joins sub-element has two attributes: type (Inner, LeftOuter, RightOuter, FullOuter, or FullOuterStitch) and cardinality (ONE TO ONE, ZERO OR ONE TO ONE, ONE TO ZERO OR ONE, ZERO OR ONE TO ZERO OR ONE, ONE TO MANY, ZERO OR ONE TO MANY, ZERO OR ONE TO MANY, MANY TO ONE, MANY TO ZERO OR ONE, or MANY TO MANY). See the syntax for the logical table source element to see the complete hierarchy for the link child element.                                      |
| WhereClause        | If a WHERE clause filter has been defined, contains the WHERE clause expression that limits the number of rows the source uses in the resultant table. The expression is contained in a sub-element called Expr.  
  In the Administration Tool, this option is called Use this "WHERE clause" filter to limit rows returned (exclude the "WHERE") in the Content tab of the Logical Table Source dialog. |
| GroupBy            | Contains the expression that defines whether the aggregation is grouped by logical level or by column. The expression is contained in a sub-element called Expr.  
  In the Administration Tool, this option is called Aggregation content, group by in the Content tab of the Logical Table Source dialog. |
Table 2–18  (Cont.) Logical Table Source Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FragmentContent</td>
<td>Contains the expression that defines how the source is fragmented. The expression is contained in a sub-element called Expr. This child element is used in conjunction with the canCombine attribute. In the Administration Tool, this option is called Fragmentation content in the Content tab of the Logical Table Source dialog.</td>
</tr>
</tbody>
</table>

Syntax
The syntax here includes the summary child element entry for the column mapping element. The full syntax for the column mapping element appears in Section 2.9.1, "Column Mapping."

```xml
<LogicalTableSource _
  isDistinct=""
  isSysGen=""
  canCombine=""
  isActive=""
  priorityGroup=""
  allowUnmappedTable=""
  logicalTableRef=""
  closureTableRef="">
  <Description />
  <ColumnMapping />
  <Link>
    startNodeTableRef=""
    <Join
      type=""
      cardinality=""
      physicalTable1Ref=""
      physicalTable2Ref=""
      physicalJoinRef="" />
  </Link>
  <WhereClause />
  <GroupBy />
  <FragmentContent />
</LogicalTableSource>
```

**Note:** Each of the summary syntax elements `<WhereClause />`, `<GroupBy />`, and `<Fragment Content />` have the following sub-elements:

```xml
<ExprText>...</ExprText>
<ExprTextDesc>...</ExprTextDesc>
<ObjectRefList>
  <RefObject objectRef="" ...
  <RefObject objectRef="" ...
</ObjectRefList>
```

Example
```xml
<LogicalTableSource mdsid="m80e13d0b-07f5-0000-714b-e31d00000000"
  name="Fact_W_REVN_F_Team_Security_Resource"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  isActive="true"
```
logicalTableRef="/oracle/bi/server/base/LogicalTable/Dim _u0150 Security _u0150 Opportunity_80e13d09-07f3-0000-714b-e31d00000000.xml#m80e13d09-07f3-0000-714b-e31d00000000">
<ColumnMapping>
  <LogicalColumnExpr>
    <ExprText>
      <![CDATA[ "%1" ]]>  
    </ExprText>
    <ExprTextDesc>
      <![CDATA[ "Core"."Dim - Security - Opportunity"."Row Wid" ]]>  
    </ExprTextDesc>
    <ObjectRefList>
      <RefObject refId="m80e13d0b-07f5-0000-714b-e31d00000000-m80e13d0d-07d6-0000-714b-e31d00000000" objectTypeId="2006" objectRef="/oracle/bi/server/base/LogicalTable/Dim _u0150 Security _u0150 Opportunity80e13d09-07f3-0000-714b-e31d00000000.xml#m80e13d0d-07d6-0000-714b-e31d00000000" />
    </ObjectRefList>
  </LogicalColumnExpr>
  <Expr>
    <ExprText>
      <![CDATA[ "%1" ]]>  
    </ExprText>
    <ExprTextDesc>
      <![CDATA[ "Oracle Data Warehouse"."Catalog"."dbo"."Fact_W_REVN_F"."ROW_WID" ]]>  
    </ExprTextDesc>
    <ObjectRefList>
      <RefObject refId="m80e13d0b-07f5-0000-714b-e31d00000000-m00228792-0bbb-0000-714b-e31d00000000" objectTypeId="3003" objectRef="/oracle/bi/server/base/PhysicalTable/Fact_W_REVN_F_0022613e-0bb9-0000-714b-e31d00000000.xml#m00228792-0bbb-0000-714b-e31d00000000" />
    </ObjectRefList>
  </Expr>
</ColumnMapping>

...
2.9.1 Column Mapping

The column mapping element does not correspond to any metadata repository object. It is only used in conjunction with the logical table source element. The column mapping element appears on its own in the MDS XML schema (.xsd file), rather than as part of the logical table source element, so that the column mapping details are easier to read.

The column mapping tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogicalColumnExpr</td>
<td>Contains the expression that identifies the logical column for this column mapping.</td>
</tr>
<tr>
<td>Expr</td>
<td>Contains the expression that identifies the physical column for this column mapping.</td>
</tr>
</tbody>
</table>

The column mapping tag does not include any attributes.

Syntax

```
<ColumnMapping>
  <LogicalColumnExpr />
  <Expr />
</ColumnMapping>
```
Example
<ColumnMapping>
  <LogicalColumnExpr>
    <ExprText>
      <![CDATA[ "%1" ]]>
    </ExprText>
    <ExprTextDesc>
      <![CDATA[ "Core"."Fact - CRM - Employee"."# Employees" ]]
    </ExprTextDesc>
    <ObjectRefList>
      <RefObject objectRef="" />
      <RefObject objectRef="" />
    </ObjectRefList>
  </LogicalColumnExpr>
  <Expr>
    <ExprText>
      <![CDATA[ "%1" ]]>
    </ExprText>
    <ExprTextDesc>
      <![CDATA[ "Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_EMPLOYEE_D"."ROW_WID" ]]
    </ExprTextDesc>
    <ObjectRefList>
      <RefObject objectRef="" />
      <RefObject objectRef="" />
    </ObjectRefList>
  </Expr>
</ColumnMapping>

2.10 MDS XML Elements: Logical Source Folder

The logical source folder element corresponds to the logical source folder object in the Business Model and Mapping layer. Logical source folders contain logical table sources.

The logical source folder tag includes the following attributes:


### 2.11 MDS XML Elements: Measure Definition

The measure definition element stores the measure definition for a measure column. A measure column is usually in a fact table, can change for each record, and can be added up or aggregated in some way.

The measure definition element is not a top-level element - it appears as a child element of the logical column element.

The measure definition tag includes the following attributes:

#### Table 2–21 Measure Definition Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isCommutative</td>
<td>If set to true, indicates that the aggregation rule lets the server switch the order if it can increase the performance. For example, the following aggregation function is commutative: ( A \ (B \ (x)) = B \ (A \ (x)) )</td>
</tr>
</tbody>
</table>

The measure definition tag includes the following child elements:

#### Table 2–22 Measure Definition Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AggrRule</td>
<td>References the aggregation rule for this measure. For details, see Section 2.12, &quot;MDS XML Elements: Aggregation Rule.&quot;</td>
</tr>
<tr>
<td>OverrideAggrRule</td>
<td>References the override aggregation rule for this measure, if any. For details, see Section 2.13, &quot;MDS XML Elements: Override Aggregation Rule.&quot;</td>
</tr>
</tbody>
</table>

#### Syntax

```xml
<MeasureDefn ...
    isCommutative="true">
    <AggrRule />
    <OverrideAggrRule />
</MeasureDefn>
```
2.12 MDS XML Elements: Aggregation Rule

The aggregation rule element corresponds to the aggregation rule object in the Business Model and Mapping layer. The aggregation rule element is not a top-level element - it appears as a child element of the measure definition element. The aggregation rule tag includes the following attributes:

**Table 2–23 Aggregation Rule Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDefault</td>
<td>If set to true, indicates that this is the default rule for aggregate calculations. This option only applies when there is more than one aggregation rule.</td>
</tr>
<tr>
<td>basedOnDimensionRef</td>
<td>If this aggregation rule is specific to a particular dimension, references the dimension for which this aggregation rule applies.</td>
</tr>
</tbody>
</table>

The aggregation rule tag includes the following child elements:

**Table 2–24 Aggregation Rule Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expr</td>
<td>Contains the expression on which this aggregation rule is based.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<AggrRule
  isDefault=""
```
2.13 MDS XML Elements: Override Aggregation Rule

The override aggregation rule element corresponds to the override aggregation expression for a logical table source in the Business Model and Mapping Layer.

When you select Count Distinct as the default aggregation rule, you can specify an override aggregation expression for specific logical table sources. This option is useful when you have more than one logical table source mapped to a logical column and you want to apply a different aggregation rule to each source.

The override aggregation rule element is not a top-level element - it appears as a child element of the measure definition element.

The override aggregation rule tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logicalTableSourceRef</td>
<td>References the logical table source for this override aggregation rule.</td>
</tr>
</tbody>
</table>

The override aggregation rule tag includes the following child elements:
### Syntax

```xml
<OverrideAggrRule _logicalTableSourceRef="">
  <Description />
  <Expr>
    <ExprText>...</ExprText>
    <ExprTextDesc>...</ExprTextDesc>
    <ObjectRefList>
      <RefObject objectRef="" />
      <RefObject objectRef="" />
    </ObjectRefList>
  </Expr>
</OverrideAggrRule>
```

### Example

```xml
<OverrideAggrRule mdsid="m80c95c48-0806-0000-714b-e31d00000000" name="OvAggRule_2054:385771246426812"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.oracle.com/obis/repository"
      logicalTableSourceRef="/oracle/bi/server/base/LogicalTableSource/Fact_Agg_W_RESP_CMPORF_A_8033657e-07f5-0000-714b-e31d00000000.xml#m8033657e-07f5-0000-714b-e31d00000000">
  <Expr>
    <ExprText>
      <![[CDATA[ SUM("%1") ]]>}
    </ExprText>
    <ExprTextDesc>
      <![CDATA[ SUM("Core"."Fact - CRM - Response"."# Responses") ]]>}
    </ExprTextDesc>
    <ObjectRefList>
      <RefObject refId="m80c95c48-0806-0000-714b-e31d00000000-m00222110-07d6-0000-714b-e31d00000000" objectTypeId='2006' objectRef="/oracle/bi/server/base/LogicalTable/Fact Response_00224ffb-07f3-0000-714b-e31d00000000.xml#m00222110-07d6-0000-714b-e31d00000000"/>
    </ObjectRefList>
  </Expr>
</OverrideAggrRule>
```

### 2.14 MDS XML Elements: Logical Display Folder

The logical display folder element corresponds to the logical display folder object in the Business Model and Mapping layer.

The logical display folder tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subjectAreaRef</td>
<td>References the business model for this logical display folder.</td>
</tr>
</tbody>
</table>

---

**Table 2–26 Override Aggregation Rule Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expr</td>
<td>Contains the expression on which this aggregation rule is based.</td>
</tr>
</tbody>
</table>
The logical display folder tag includes the following child elements:

**Table 2–28 Logical Display Folder Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>References the dimensions that belong to this logical display folder. Each dimension is contained in a sub-element called RefDimension.</td>
</tr>
<tr>
<td>Tables</td>
<td>References the logical tables that belong to this logical display folder. Each table is contained in a sub-element called RefLogicalTable.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<LogicalDisplayFolder _
    subjectAreaRef=""
    >
    <Description />
    <RefDimensions>
        <RefDimension dimensionRef="" _/>
    </RefDimensions>
    <RefTables>
        <RefLogicalTable logicalTableRef="" _/>
    </RefTables>
</LogicalDisplayFolder>
```

**Example**

```xml
<LogicalDisplayFolder mdsid="me4070ae0-0bf1-1000-b290-0ae593200000"
    name="Fact Tables"
    iconIndex="69"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    subjectAreaRef="/oracle/bi/server/base/BusinessModel/Time Series_e406baf8-0bf1-1000-b290-0ae593200000.xml#me406baf8-0bf1-1000-b290-0ae593200000">
    <RefTables>
        <RefLogicalTable refId="me4070ae0-0bf1-1000-b290-0ae593200000-me406e378-0bf1-1000-b290-0ae593200000"
            logicalTableRef="/oracle/bi/server/base/LogicalTable/SalesFacts_e406e378-0bf1-1000-b290-0ae593200000.xml#me406e378-0bf1-1000-b290-0ae593200000" />
        <RefLogicalTable refId="me4070ae0-0bf1-1000-b290-0ae593200000-me406e456-0bf1-1000-b290-0ae593200000"
            logicalTableRef="/oracle/bi/server/base/LogicalTable/SalesPlan_e406e456-0bf1-1000-b290-0ae593200000.xml#me406e456-0bf1-1000-b290-0ae593200000" />
    </RefTables>
</LogicalDisplayFolder>
```
This chapter provides MDS XML reference information for physical elements. Physical elements typically correspond to objects in the Physical layer of the Oracle BI repository, or in other words, the objects that appear in the right-hand pane of the Oracle BI Administration Tool.

This chapter contains the following topics:

- Section 3.1, "MDS XML Elements: Database"
- Section 3.2, "MDS XML Elements: Physical Catalog"
- Section 3.3, "MDS XML Elements: Physical Schema"
- Section 3.4, "MDS XML Elements: Physical Display Folder"
- Section 3.5, "MDS XML Elements: Physical Table"
- Section 3.6, "MDS XML Elements: Physical Column"
- Section 3.7, "MDS XML Elements: Connection Pool"
- Section 3.8, "MDS XML Elements: Physical Key"
- Section 3.9, "MDS XML Elements: Physical Foreign Key"
- Section 3.10, "MDS XML Elements: Complex Join"
- Section 3.11, "MDS XML Elements: Cube Table"
- Section 3.12, "MDS XML Elements: Physical Dimension"
- Section 3.13, "MDS XML Elements: Physical Hierarchy"
- Section 3.14, "MDS XML Elements: Physical Level"
- Section 3.15, "MDS XML Elements: Cube Column"
- Section 3.16, "MDS XML Elements: Analytic Workspace"
- Section 3.17, "MDS XML Elements: AW Cube"
- Section 3.18, "MDS XML Elements: AW Dimension"
- Section 3.19, "MDS XML Elements: AW Hierarchy"
- Section 3.20, "MDS XML Elements: AW Level"
- Section 3.21, "MDS XML Elements: AW Column"

3.1 MDS XML Elements: Database

The database element corresponds to the database object in the Physical layer.
The database tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of data source, such as XMLDirect or ODBC20.</td>
</tr>
<tr>
<td>dbFlag</td>
<td>Stores the value of the following options:</td>
</tr>
<tr>
<td></td>
<td>- <strong>CRM metadata tables</strong>: For legacy Siebel Systems sources only. When selected, the Administration Tool import utility looks for the table definition in Oracle's Siebel CRM-specific tables.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Virtual Private Database</strong>: Identifies the physical database source as a virtual private database (VPD). When a VPD is used, returned data results are contingent on the user's authorization credentials. This option is used with the Security Sensitive option for session variables.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Allow populate queries by default</strong>: When selected, allows all users to execute <code>POPULATE SQL</code>. If you want most, but not all, users to be able to execute <code>POPULATE SQL</code>, select this option and then limit queries for specific users or groups.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Allow direct database requests by default</strong>: When selected, allows all users to execute physical queries. The Oracle BI Server will send unprocessed, user-entered, physical SQL directly to an underlying database. If you want most, but not all, users to be able to execute physical queries, select this option and then limit queries for specific users or groups.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the container for this database.</td>
</tr>
<tr>
<td>persistConnPoolRef</td>
<td>References the connection pool that is used as the persist connection pool, if one has been assigned. A persist connection pool is a database property that is used for specific types of queries (typically used to support Marketing queries).</td>
</tr>
</tbody>
</table>

The database tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>Lists the SQL features for this database. Each feature is contained in a sub-element called <code>Feature</code>. This sub-element has two attributes:</td>
</tr>
<tr>
<td></td>
<td>- <code>name</code> indicates the name of the feature, such as <code>LEFT_OUTER_JOIN_SUPPORTED</code>.</td>
</tr>
<tr>
<td></td>
<td>- <code>value</code> indicates whether this feature is supported by the database, or provides the actual value of the feature (such as &quot;0&quot; for <code>MAX_COLUMNS_IN_SELECT</code>).</td>
</tr>
<tr>
<td>DisplayFolders</td>
<td>References any physical display folders that contain objects in this database. Physical display folders can be used to organize tables in a database. For example, all alias tables can be stored in a display folder called Alias, in addition to being present in the database.</td>
</tr>
<tr>
<td></td>
<td>Each display folder is contained in a sub-element called <code>RefPhysicalDisplayFolder</code>.</td>
</tr>
<tr>
<td>RefConnectionPools</td>
<td>References the connection pools for this database object.</td>
</tr>
<tr>
<td></td>
<td>Each connection pool is contained in a sub-element called <code>RefConnectionPool</code>.</td>
</tr>
</tbody>
</table>
Table 3–2  (Cont.) Database Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Custom database properties for Oracle ADF Business Component data sources that are passed to the Administration Tool from Oracle ADF BI view objects.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<Database type="" dbFlag="" containerRef="" persistConnectionPoolRef="">
  <Description />
  <Features>
    <Feature name="" value=""/>
    <Feature name="" value=""/>
    ...
  </Features>
  <RefConnectionPools>
    <RefConnectionPool connectionPoolRef="" />
    <RefConnectionPool connectionPoolRef="" />
  </RefConnectionPools>
  <RefDisplayFolders>
    <RefPhysicalDisplayFolder physicalDisplayFolderRef="" />
    <RefPhysicalDisplayFolder physicalDisplayFolderRef="" />
  </RefDisplayFolders>
  <Properties>
    <PropertyGroup category="">
      <Property>
        <Name "/>
        <Value "/>
      </Property>
    </PropertyGroup>
  </Properties>
</Database>
```

Example

```xml
<Database mdsid="m80c882ea-0bcf-0000-714b-e31d00000000">
  name="Loyalty Input Data Source"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  type="ODBC200">
  <Feature name="LEFT_OUTER_JOIN_SUPPORTED" value="true" />
  <Feature name="RIGHT_OUTER_JOIN_SUPPORTED" value="false" />
  <Feature name="FULL_OUTER_JOIN_SUPPORTED" value="false" />
  <Feature name="NESTED_OUTER_JOIN_SUPPORTED" value="false" />
  <Feature name="UNION_SUPPORTED" value="false" />
  <Feature name="UNION_ALL_SUPPORTED" value="false" />
  <Feature name="COUNT_SUPPORTED" value="true" />
  <Feature name="SORT_ORDER_LOCALE" value="english-usa" />
  <Feature name="COMMENT_START" />
  <Feature name="COMMENT_END" />
  <RefConnectionPools>
    <RefConnectionPool refId="m80c882ea-0bcf-0000-714b-e31d00000000-m80c882ec-0bd5-0000-714b-e31d0000000000"
      connectionPoolRef="/oracle/bi/server/base/ConnectionPool/Loyalty Input"
```

MDS XML Physical Elements  3-3
3.2 MDS XML Elements: Physical Catalog

The physical catalog element corresponds to the physical catalog object in the Physical layer.

The physical catalog tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>containerRef</td>
<td>References the database for this physical catalog.</td>
</tr>
<tr>
<td>dynamicNameRef</td>
<td>If a session variable is being used to specify the name of this physical catalog, this element references that session variable.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<PhysicalCatalog _
    containerRef=""
    dynamicNameRef=""/>
<Description /></PhysicalCatalog>
```

Example

```xml
<PhysicalCatalog mdsid="m00238422-0bd1-0000-714b-e31d00000000"
    name="Catalog"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    containerRef="/oracle/bi/server/base/Database/Forecasting Oracle Data Warehouse
    _00238415-0bcf-0000-714b-e31d00000000.xml#m00238415-0bcf-0000-714b-
    e31d00000000"/>
</PhysicalCatalog>
```

3.3 MDS XML Elements: Physical Schema

The physical schema element corresponds to the physical schema object in the Physical layer.

The physical schema tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>containerRef</td>
<td>References the database for this physical schema.</td>
</tr>
<tr>
<td>dynamicNameRef</td>
<td>If a session variable is being used to specify the name of this physical schema, this element references that session variable.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<Schema _>
    containerRef=""
    dynamicNameRef=""/>
<Description /></Schema>
```
3.4 MDS XML Elements: Physical Display Folder

The physical display folder element corresponds to the physical display folder object in the Physical layer.

The physical display folder tag includes the following attributes:

**Table 3–5 Physical Display Folder Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>databaseRef</td>
<td>References the database for this physical display folder.</td>
</tr>
</tbody>
</table>

The physical display folder tag includes the following child elements:

**Table 3–6 Physical Display Folder Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefTables</td>
<td>References the physical tables that belong to this physical display folder. Each physical table is contained in a sub-element called RefPhysicalTable.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<PhysicalDispFolder ...>
    <databaseRef=''></databaseRef>
    <Description />
    <RefTables>
        <RefPhysicalTable tableRef=''></RefPhysicalTable>
        <RefPhysicalTable tableRef=''></RefPhysicalTable>
    </RefTables>
</PhysicalDispFolder>
```

**Example**

```xml
<PhysicalDisplayFolder mdsid='me40757el-0bf1-1000-b290-0ae593200000' name='Opaque Views' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns='http://www.oracle.com/obis/repository' databaseRef='oracle/bi/server/base/Database/SQldb_Northwind_e4070ae5-0bf1-1000-b290-0ae593200000.xml#me4070ae5-0bf1-1000-b290-0ae593200000'>
    <RefTables>
        <RefPhysicalTable refId='me40757el-0bf1-1000-b290-0ae593200000-me4072f9e-0bf1-1000-b290-0ae593200000' tableRef='oracle/bi/server/base/PhysicalTable/Employees_Country_e4072f9e-0bf1-1000-b290-0ae593200000.xml#me4072f9e-0bf1-1000-b290-0ae593200000' />
        <RefPhysicalTable refId='me40757el-0bf1-1000-b290-0ae593200000-me4072fc7-0bf1-1000-b290-0ae593200000' tableRef='oracle/bi/server/base/PhysicalTable/Population_e4072fc7-0bf1-1000-b290-0ae593200000.xml#me4072fc7-0bf1-1000-b290-0ae593200000' />
    </RefTables>
</PhysicalDisplayFolder>
```
3.5 MDS XML Elements: Physical Table

The physical table element corresponds to the physical table object in the Physical layer.

The physical table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| type | The type of physical table. Options are as follows:  
  - select indicates that this physical table object is a SELECT statement.  
  - sp indicates that this physical table is a stored procedure.  
  - alias indicates that this physical table is an alias table. An alias table (alias) is a physical table that references a different physical table as its source.  
  - none indicates that this physical table is a regular physical table. |
| isCacheable | If set to true, indicates that this table is included in the Oracle BI Server query cache. |
| CacheExpiry | How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire. |
| isCacheEventTable | If set to true, indicates that this table has been registered as an Oracle BI Server event polling table. |
| isOverrideCacheProp | For alias tables only. If set to true, indicates that the alias table has its own cache properties that override the cache properties of the source table. |
| pollFreq | The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds. |
| x | The x coordinate of the table when it is shown in the physical diagram. |
| y | The y coordinate of the table when it is shown in the physical diagram. |
| rowCount | The latest number of rows retrieved for this physical table. |
| lastUpdated | The time at which row counts were last updated for this physical table. |
| uri | For XML data sources only. The resource identifier for the XML data source. |
The physical table tag includes the following child elements:

Table 3–7 (Cont.) Physical Table Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xslPath</td>
<td>For XML data sources only. An XSLT file contains formatting rules written according to the XSLT standard. It defines how an XML file may be transformed.</td>
</tr>
<tr>
<td>maxConn</td>
<td>For XML data sources only. The maximum number of connections allowed to the same URL from different users.</td>
</tr>
<tr>
<td>xmlRefreshInterval</td>
<td>For XML data sources only. The time interval after which the XML data source is queried again directly, rather than using results in cache.</td>
</tr>
<tr>
<td>scriptPath</td>
<td>For XML data sources only. The extended gateway script path.</td>
</tr>
<tr>
<td>hints</td>
<td>For Oracle Databases only. Lists any database hints, such as index hints or leading hints, that were specified for this physical table.</td>
</tr>
<tr>
<td>isExtGatewayInProcess</td>
<td>For XML data sources only. Indicates that this physical table is the extended gateway part of the process.</td>
</tr>
<tr>
<td>extGatewayDelimiter</td>
<td>For XML data sources only. The row delimiter used in the metadata import process, typically the second level element in the XML document.</td>
</tr>
<tr>
<td>isSysGen</td>
<td>If set to true, indicates that an internal utility has generated this physical table.</td>
</tr>
<tr>
<td>isDDLAggr</td>
<td>Indicates whether this physical table is an aggregate table.</td>
</tr>
<tr>
<td>isDeployedView</td>
<td>For relational databases only. If set to true, indicates that this physical table is a deployed (opaque) view. A physical layer table consists of a SELECT statement.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the database, schema, or physical catalog for this physical table.</td>
</tr>
<tr>
<td>aggrConnPoolRef</td>
<td>If this physical table is an aggregate table, this element references the connection pool used for the aggregate.</td>
</tr>
<tr>
<td>sourceTableRef</td>
<td>For alias tables only. References the physical table that is being used as the source for this alias table.</td>
</tr>
<tr>
<td>variableRef</td>
<td>If a session variable is being used to specify the name of this physical table, this element references that session variable.</td>
</tr>
<tr>
<td>memberKeyRef</td>
<td>Applies to parent-child relationship tables that support parent-child hierarchies on relational sources. The name of the column that uniquely identifies members of the parent-child relationship table.</td>
</tr>
<tr>
<td>parentKeyRef</td>
<td>The name of the column that contains the key of the parent or ancestor member. Applies to parent-child relationship tables that support parent-child hierarchies on relational sources.</td>
</tr>
<tr>
<td>distanceColumnRef</td>
<td>The name of the column that records the number of levels between the member and the ancestor. Applies to parent-child relationship tables that support parent-child hierarchies on relational sources.</td>
</tr>
<tr>
<td>leafColumnRef</td>
<td>The name of the column that indicates whether the member has no child members (1 if the member is a leaf and has no children, 0 if the member is not a leaf and has children). Applies to parent-child relationship tables that support parent-child hierarchies on relational sources.</td>
</tr>
</tbody>
</table>
### MDS XML Elements: Physical Table

#### Table 3–8 Physical Table Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMapItem</td>
<td>Stores the value of database-specific strings. The <code>&lt;DBMapItem&gt;</code> element contains a <code>&lt;name&gt;</code> attribute and a <code>&lt;Value&gt;</code> sub-element.</td>
</tr>
<tr>
<td>PhysicalColumn</td>
<td>Specifies the physical column that belongs to this physical table. For details, see Section 3.6, &quot;MDS XML Elements: Physical Column.&quot;</td>
</tr>
<tr>
<td>PhysicalKey</td>
<td>References the physical columns in this physical table that make up the physical key. Each physical column is referenced in a sub-element called <code>RefColumns</code>. For details, see Section 3.8, &quot;MDS XML Elements: Physical Key.&quot;</td>
</tr>
<tr>
<td>PhysicalForeignKey</td>
<td>References the physical key in an external physical table and the physical columns in this physical table that together make up the physical foreign key. Each physical column is referenced in a sub-element called <code>RefColumns</code>. For details, see Section 3.9, &quot;MDS XML Elements: Physical Foreign Key.&quot;</td>
</tr>
<tr>
<td>XSLT</td>
<td>For Oracle web service data sources. XSLT stores the XSLT transformation to flatten the web service response into a tabular result set.</td>
</tr>
<tr>
<td>XSDSchema</td>
<td>For Oracle web service data sources. XSDSchema stores the XML schema describing the structure of the web service response.</td>
</tr>
<tr>
<td>Properties</td>
<td>Contains one or more <code>Property</code> sub-elements, each of which has attributes <code>&lt;name&gt;</code> and <code>&lt;value&gt;</code>. These attributes specify a name-value pair of properties for data source metadata.</td>
</tr>
</tbody>
</table>

### Syntax

The syntax here includes summary child element entries for the physical column, physical key, and physical foreign key elements. The full syntax for each child element appears in the appropriate section listed in the following table:

#### Table 3–9 References to Child Elements of Physical Table Element

<table>
<thead>
<tr>
<th>Child Element Entry</th>
<th>Section Containing Child Element Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhysicalColumn</td>
<td>MDS XML Elements: Physical Column</td>
</tr>
<tr>
<td>PhysicalKey</td>
<td>MDS XML Elements: Physical Key</td>
</tr>
<tr>
<td>PhysicalForeignKey</td>
<td>MDS XML Elements: Physical Foreign Key</td>
</tr>
</tbody>
</table>

```xml
<PhysicalTable ...
  type=""
  isCacheable=""
  cacheExpiry=""
  isCacheEventTable=""
  isOverrideCacheProp=""
  pollFreq=""
  x=""
  y=""
  rowCount=""
  lastUpdated=""
  uri=""
```
Example

```xml
<PhysicalTable mdsid="m00226232-0bb9-0000-714b-e31d00000000"
    name="CONTACTS_PRODUCTS_NOT_ORDERED"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    type="select"
    x="625"
    y="376"
    maxConn="0"
    containerRef="/oracle/bi/server/base/Schema/dbo_00238426-0bd3-0000-714b-
e31d00000000.xml#m00238426-0bd3-0000-714b-e31d00000000">
    <DBMapItem name="SQLServer70">
        <Value><![CDATA[
            select distinct T10.ROW_WID AS CONTACT_WID, T11.ROW_WID AS PRODUCT_WID
            from VALUEOF(OLAPTBO).W_PERSON_D T10, VALUEOF(OLAPTBO).W_PRODUCT_D T11
            where Not EXISTS (
                select * from VALUEOF(OLAPTBO).W_ORDERITEM_F T1 where T1.CONTACT_WID = T10.ROW_WID and T1.PROD_WID = T11.ROW_WID)
        ]]>]
    </DBMapItem>
    <DBMapItem name="DefaultMulDB">
        <Value><![CDATA[
            select distinct T10.ROW_WID AS CONTACT_WID, T11.ROW_WID AS PRODUCT_WID
            from VALUEOF(OLAPTBO).W_PERSON_D T10, VALUEOF(OLAPTBO).W_PRODUCT_D T11
            where Not EXISTS {
                select * from VALUEOF(OLAPTBO).W_ORDERITEM_F T1 where T1.CONTACT_WID = T10.ROW_WID and T1.PROD_WID = T11.ROW_WID
            }
        ]]>]
    </DBMapItem>
</PhysicalTable>
```
select * from VALUEOF (OLAPTBO).W_ORDERITEM_F T1 where T1.CONTACT_WID = T10.ROW_WID and T1.PROD_WID = T11.ROW_WID)
}>
</Value>
</DBMapItem>
<PhysicalColumn mdsid="m0022ba68-0bb-0000-714b-e31d00000000"
name="CONTACT_WID"
dataType="INT" specialType="none">
</PhysicalColumn>
<PhysicalColumn mdsid="m0022ba69-0bb-0000-714b-e31d00000000"
name="PRODUCT_WID"
dataType="INT" specialType="none">
</PhysicalColumn>
<PhysicalKey mdsid="m00238038-0bc-0000-714b-e31d00000000"
name="CONTACT_PRODUCT_KEY">
<RefColumns>
<RefColumn refId="m00238038-0bc-0000-714b-e31d00000000-m0022ba68-0bb-0000-714b-e31d00000000"
columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba68-0bb-0000-714b-e31d00000000" />
<RefColumn refId="m00238038-0bc-0000-714b-e31d00000000-m0022ba69-0bb-0000-714b-e31d00000000"
columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba69-0bb-0000-714b-e31d00000000" />
</RefColumns>
</PhysicalKey>
<PhysicalForeignKey mdsid="m002377f8-0bbe-0000-714b-e31d00000000"
name="CONTACTS_PRODUCTS_NOT_ORDERED_FKey" counterPartKeyRef="/oracle/bi/server/base/PhysicalTable/Dim_W_PARTY_PER_D_CContact_002261ab-0bb9-0000-714b-e31d00000000.xml#m00237fa7-0bc-0000-714b-e31d00000000"">
<RefColumns>
<RefColumn refId="m002377f8-0bbe-0000-714b-e31d00000000-m0022ba68-0bb-0000-714b-e31d00000000"
columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba68-0bb-0000-714b-e31d00000000" />
<RefColumn refId="m002377f8-0bbe-0000-714b-e31d00000000-m0022ba69-0bb-0000-714b-e31d00000000"
columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba69-0bb-0000-714b-e31d00000000" />
</RefColumns>
</PhysicalForeignKey>
<PhysicalForeignKey mdsid="m002377f9-0bbe-0000-714b-e31d00000000"
name="CONTACTS_PRODUCTS_NOT_ORDERED_FKey1" counterPartKeyRef="/oracle/bi/server/base/PhysicalTable/Dim_W_PRODUCT_D_0022612d-0bb9-0000-714b-e31d00000000.xml#m00237f18-0bc-0000-714b-e31d00000000"">
<RefColumns>
<RefColumn refId="m002377f9-0bbe-0000-714b-e31d00000000-m0022ba68-0bb-0000-714b-e31d00000000"
columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba68-0bb-0000-714b-e31d00000000" />
<RefColumn refId="m002377f9-0bbe-0000-714b-e31d00000000-m0022ba69-0bb-0000-714b-e31d00000000"
columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba69-0bb-0000-714b-e31d00000000" />
</RefColumns>
</PhysicalForeignKey>
</PhysicalTable>
3.6 MDS XML Elements: Physical Column

The physical column element corresponds to the physical column object in the Physical layer.

The physical column element is not a top-level element - it appears as a child element of the physical table element.

The physical column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The external name for the physical column. This attribute is required if the same name (such as STATE) is used in multiple hierarchies. This attribute is optional for XML documents. For XML, the extName attribute stores the fully qualified name of a column.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the physical column, such as VARCHAR. This option is called Type in the Physical Column dialog in the Administration Tool.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the column. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the column.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the column.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this physical column.</td>
</tr>
<tr>
<td>lastUpdated</td>
<td>The time at which row counts were last updated for this physical column.</td>
</tr>
<tr>
<td>valueType</td>
<td>Applies to Siebel OM data sources only. Indicates the cardinality of the value of the column. Valid values include: \n- One to One (Default/Unknown/Single-valued) \n- Many to One (Picklist) \n- One to Many (Multi-valued) \n- Many to Many (Multi-valued)</td>
</tr>
<tr>
<td>mappedColumn</td>
<td>Applies to Siebel OM data sources only. The name of the mapped column. Corresponds to the field of the same name for the multi-valued group (MVG) business component.</td>
</tr>
<tr>
<td>MVLLinkName</td>
<td>Applies to Siebel OM data sources only. The name of the link used in the Siebel business component.</td>
</tr>
<tr>
<td>specialType</td>
<td>Valid values are LUD or None. LUD stands for “Last Update Date.” This attribute is currently used only for ADF data sources. It is used to indicate that the column should be checked when performing ETL operations.</td>
</tr>
<tr>
<td>dynamicNameRef</td>
<td>If a session variable is being used to specify the name of this physical column, this element references that session variable.</td>
</tr>
<tr>
<td>sourceColumnRef</td>
<td>For physical columns in an alias table only. References the physical column that is being used as the source for this column.</td>
</tr>
</tbody>
</table>

The physical column tag includes the following child elements:
Table 3–11  Physical Column Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Contains one or more Property sub-elements, each of which has attributes &lt;name&gt; and &lt;value&gt;. These attributes specify a name-value pair of properties for data source metadata.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<PhysicalColumn _
    extName=""
dataType=""
    length=""
    nullable=""
    precision=""
    scale=""
    rowCount=""
    lastUpdated=""
    valueType=""
    mappedColumn=""
    MVLinkName=""
    specialType=""
    dynamicNameRef=""
    sourceColumnRef=''>
    <Description />
    <Properties>
        <Property name=''' value='''/>
        <Property name=''' value='''/>
    </Properties>
</PhysicalColumn>
```

Example

```xml
<PhysicalColumn mdsid='m80e86ae3-0bbb-0000-714b-e31d00000000'
    name='ReportDate'
dataType='TIMESTAMP' scale='127' specialType='none'
    sourceColumnRef='''oracle/bi/server/base/PhysicalTable/FscmTopModelAM.Gregorian
    CalendarAM.FndCa1Day_80d8ff44-0bb9-0000-714b-e31d00000000.xml#m80d759cf-0bbb-
    0000-714b-e31d00000000'>
</PhysicalColumn>
```

3.7 MDS XML Elements: Connection Pool

The connection pool element corresponds to the connection pool object in the Physical layer.

The connection pool tag includes the following attributes:

Table 3–12  Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bulkInsertBufferSize</td>
<td>Used for limiting the number of bytes each time data is inserted in a database table. The value is given in kilobytes (KB).</td>
</tr>
<tr>
<td>transactionBoundary</td>
<td>Controls the batch size for an insert in a database table.</td>
</tr>
<tr>
<td>unicodeDB</td>
<td>This attribute is typically set to true when working with columns of an explicit Unicode data type, such as NCHAR, in an Unicode database.</td>
</tr>
</tbody>
</table>
### Table 3–12  (Cont.) Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tempTablePrefix</td>
<td>The first two characters in a temporary table name. The default value is TT.</td>
</tr>
<tr>
<td>tempTableOwner</td>
<td>Table owner name used to qualify a temporary table name in a SQL statement, for example to create the table owner.tablename.</td>
</tr>
<tr>
<td>tempTableDatabase</td>
<td>Database where the temporary table will be created. This property applies only to IBM OS/390, because IBM OS/390 requires the database name qualifier to be part of the CREATE TABLE statement.</td>
</tr>
<tr>
<td>tempTableSpace</td>
<td>Tablespace where the temporary table will be created. This property applies only to IBM OS/390, because IBM OS/390 requires the tablespace name qualifier to be part of the CREATE TABLE statement.</td>
</tr>
<tr>
<td>user</td>
<td>The user name for the data source.</td>
</tr>
<tr>
<td>password</td>
<td>The password for the data source. Note that passwords are encrypted.</td>
</tr>
<tr>
<td>timeout</td>
<td>The amount of time that a connection to the data source will remain open after a request completes. During this time, new requests use this connection rather than open a new one (up to the number specified for the maximum connections). The time is reset after each completed connection request.</td>
</tr>
<tr>
<td>maxConnDiff</td>
<td>The maximum number of total connections allowed to this connection pool across all users.</td>
</tr>
<tr>
<td>dataSource</td>
<td>The data source name that is configured to access the database to which you want to connect. The data source name must contain valid logon information for the data source.</td>
</tr>
<tr>
<td>reqQualifiedTableName</td>
<td>If set to true, indicates that this database requires fully qualified table names. The fully qualified names are based on the physical object names in the repository.</td>
</tr>
<tr>
<td>isSharedLogin</td>
<td>If set to true, all users whose queries use the connection pool will access the underlying database using the same user name and password.</td>
</tr>
<tr>
<td>isConcurrentQueriesInConnection</td>
<td>An option that enables a single database connection to remain open for the specified time for use by future query requests. If you do not select this option, each query sent to the database opens a new connection.</td>
</tr>
<tr>
<td></td>
<td>In the Administration Tool, this option corresponds to a value of “selected” for Enable connection pooling in the General tab of the Connection Pool dialog.</td>
</tr>
<tr>
<td>isCloseAfterEveryRequest</td>
<td>If set to true, indicates that the connection pool gets closed after every query.</td>
</tr>
<tr>
<td></td>
<td>In the Administration Tool, this option corresponds to a value of “not selected” for Enable connection pooling in the General tab of the Connection Pool dialog.</td>
</tr>
<tr>
<td>isAsyncExecution</td>
<td>An option that indicates that the data source supports asynchronous queries.</td>
</tr>
</tbody>
</table>
### Table 3–12  (Cont.) Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>executeOnCmd</td>
<td>A command that is to be executed by the Oracle BI Server each time a connection is made to the database. The command may be any command accepted by the database.</td>
</tr>
<tr>
<td>xslPath</td>
<td>For XML data sources only. An XSLT file contains formatting rules written according to the XSLT standard. It defines how an XML file may be transformed. Specifying the XSLT file in the connection pool applies it to all the XML physical tables in the connection pool. Used when outputType is set to XML.</td>
</tr>
<tr>
<td>maxConn</td>
<td>The maximum number of total connections allowed to this connection pool for a given user.</td>
</tr>
<tr>
<td>xmlRefreshInterval</td>
<td>For XML data sources only. The time interval after which the XML data source is queried again directly, rather than using results in cache.</td>
</tr>
<tr>
<td>scriptPath</td>
<td>The connection method Search script. This option is used for XML Server data sources.</td>
</tr>
<tr>
<td>outputType</td>
<td>For XML data sources only. The query output format (delimited, xml, or fixed).</td>
</tr>
<tr>
<td></td>
<td>In the Administration Tool, this option is called <strong>Query output format</strong> in the XML tab of the Connection Pool dialog.</td>
</tr>
<tr>
<td></td>
<td>If this option is not set, the query output format is assumed to be a custom format, and the value for <code>convertScriptPath</code> is used.</td>
</tr>
<tr>
<td>gwDelim</td>
<td>The delimiting characters for the query output format. Used when outputType is set to delimited.</td>
</tr>
<tr>
<td>xpath</td>
<td>An XPath expression is a simple XSLT transformation rule that fits into one line. Used when outputType is set to xml.</td>
</tr>
<tr>
<td>convertScriptPath</td>
<td>Used when outputType is not set. The path to the .exe file to be used to convert the query output to a custom output format (such as HTML).</td>
</tr>
<tr>
<td>ignoreFirstLine</td>
<td>If set to true, indicates that the first line of the output file is ignored.</td>
</tr>
<tr>
<td>xmlHeaderPath</td>
<td>The path to the header file, if any, used as a query input supplement. Used for XML Server data sources.</td>
</tr>
<tr>
<td>trailerPath</td>
<td>The path to the trailer file, if any, used as a query input supplement. Used for XML Server data sources.</td>
</tr>
<tr>
<td>supportParams</td>
<td>Indicates whether the data source supports parameters. If set to false, and the database features table supports parameters, special code executes that enables the Oracle BI Server to push filters (or calculations) with parameters to the database.</td>
</tr>
</tbody>
</table>
### Table 3–12 (Cont.) Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| isolationLevel | For ODBC and DB2 gateways. Sets the transaction isolation level on each connection to the back-end database. The isolation level setting controls the default transaction locking behavior for all statements issued by a connection. Options are as follows:  
  - default uses the default transaction locking behavior of the data source.  
  - readCommitted specifies that shared locks are held while the data is read to avoid dirty reads.  
  - readUncommitted implements dirty read (isolation level 0 locking). When this option is set, it is possible to read uncommitted or dirty data, change values in the data, and have rows appear or disappear in the data set before the end of the transaction.  
  - repeatableRead places locks on all data that is used in a query, preventing other users from updating the data.  
  - serializable places a range lock on the data set, preventing other users from updating or inserting rows into the data set until the transaction is complete.  |
| xmlaInfo       | The vendor-specific information used to connect to the multidimensional data source. |
| xmlaCatalog    | The list of catalogs available. The cube tables correspond to the catalog you use in the connection pool. |
| xmlaUseSession | If set to true, causes queries go through a common session.                   |
| multiThreaded  | If set to true, indicates that the connection pool supports multi-threading, or in other words, that one connection pool can support multiple queries. If this option is set to false, each query is tied to a single database connection. |
| type           | The call interface type.                                                    |
| appServerName  | The name of the ADF application server.                                     |
| appServerPort  | The port number of the ADF application server.                             |
| appID          | The ADF application ID.                                                     |
| appEJBBean     | The name of the ADF application Enterprise Java Bean.                       |
| javaHostServiceName | The name of the JavaHost service. The JavaHost is used as a connector between the Oracle BI Server and data sources that use Java, such as ADF. |
| isSiebelJDBSecured | For Siebel OM data sources only. Indicates whether the connection between the Oracle BI Server and the Siebel data source is encrypted. |
| csfKey         | This attribute is not currently used.                                       |
| isAW           | This attribute is not currently used.                                       |
| useCert        | This attribute stands for “Use Certificate.” It is used for ADF data sources and indicates whether to use a certificate when securing the connection. |
| wsServer       | Specifies the Oracle database web service host name.                        |
| wsWSDL         | Specifies the WSDL URL to describe the Oracle database web service.        |
| databaseRef    | References the database for this connection pool.                          |
The connection pool tag includes the following child elements:

<table>
<thead>
<tr>
<th>Table 3–13 Connection Pool Child Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element Name</strong></td>
</tr>
<tr>
<td>AfterQueryString</td>
</tr>
<tr>
<td>BeforeQueryString</td>
</tr>
<tr>
<td>PostConnectString</td>
</tr>
<tr>
<td>PreDisconnectString</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<ConnectionPool _
    bulkInsertBufferSize=""
    transactionBoundary=""
    unicodeDB=""
    tempTablePrefix=""
    tempTableOwner=""
    tempTableDatabase=""
    tempTableSpace=""
    user=""
    password=""
    timeout=""
    maxConnDiff=""
    dataSource=""
    reqQualifedTableName=""
    isSharedLogin=""
    isConcurrentQueriesInConnection=""
    isCloseAfterEveryRequest=""
    isAsyncExecution=""
    executeOnCmd=""
    xslPath=""
    maxConn=""
    xmlRefreshInterval=""
    scriptPath=""
    outputType=""
    gwDelim=""
    xpath=""
    convertScriptPath=""
    ignoreFirstLine=""
    xmlHeaderPath=""
    trailerPath=""
```

---

Table 3–12 (Cont.) Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>relationalConnectionPoolRef</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>relationalDataFolderRef</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>sqlBypassDBRef</td>
<td>For Oracle ADF data sources only. References the SQL Bypass database, if one has been specified. The SQL Bypass feature directly queries the database so that aggregations and other transformations are pushed down where possible, reducing the amount of data streamed and worked on in Oracle Business Intelligence.</td>
</tr>
</tbody>
</table>

---

The SQL Bypass feature directly queries the database so that aggregations and other transformations are pushed down where possible, reducing the amount of data streamed and worked on in Oracle Business Intelligence.
Example

<ConnectionPool mdsid="m80b43a95-0bd5-0000-714b-e31d00000000"
  name="Oracle EBS OLTP Connection Pool"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  user="VALUEOF(ORA_EBS_OLTP_USER)"
  password="6C6F08E01FC4111F3AC2236A364D912B0284EB262305E146621D44DB80DC27D4E24807068A0EA49C11EFFFFE9741BB7046FB2852F64FB86F0B378C4012BF5A56769A9CF97FA4051F08C9679B3C5486E"
  timeout="300"
  maxConnDiff='10'
  maxConn='64'
  dataSource='VALUEOF(ORA_EBS_OLTP_DSN)'
  type='Default'
  reqQualifiedTableName='false'
  isSharedLogin='true'
  isConcurrentQueriesInConnection='false'
  isCloseAfterEveryRequest='false'
  outputType='xml'
  ignoreFirstLine='false'
  bulkInsertBufferSize='32768'
  tempTablePrefix='TT'
  transactionBoundary='10'
  xmlaUseSession='false'
  isolationLevel='readUncommitted'
  isSiebelJDBSecured='false'
3.8 MDS XML Elements: Physical Key

The physical key element corresponds to the physical key object in the Physical layer. The physical key element is not a top-level element - it appears as a child element of the physical table element.

The physical key tag includes the following child elements:

Table 3–14 Physical Key Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefColumns</td>
<td>References the column that defines this physical key. The column is contained in a sub-element, either RefPhysicalColumn or RefCubeColumn.</td>
</tr>
</tbody>
</table>

The physical key element does not include any attributes.

Syntax

```xml
<PhysicalKey _>
  <Description />
  <RefColumns>
    <RefColumn columnRef="" _/>
  </RefColumns>
</PhysicalKey>
```

Example

```xml
<PhysicalKey mdsid="m00238038-0bc0-0000-714b-e31d00000000" name="CONTACT_PRODUCT_KEY">
  <RefColumns>
    <RefColumn refId="m00238038-0bc0-0000-714b-e31d00000000-m0022ba68-0bbe-0000-714b-e31d00000000" columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba68-0bbe-0000-714b-e31d00000000" />
    <RefColumn refId="m00238038-0bc0-0000-714b-e31d00000000-m0022ba69-0bbb-0000-714b-e31d00000000" columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_ORDERED_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba69-0bbb-0000-714b-e31d00000000" />
  </RefColumns>
</PhysicalKey>
```
3.9 MDS XML Elements: Physical Foreign Key

The physical foreign key element corresponds to the physical foreign key object in the Physical layer.

The physical foreign key element is not a top-level element - it appears as a child element of the physical table element.

The physical foreign key tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hints</td>
<td>For Oracle Databases only. Lists any database hints, such as index hints or leading hints, that were specified for this physical foreign key.</td>
</tr>
<tr>
<td>counterPartKeyRef</td>
<td>References the physical key in the primary key table, that matches the RefColumns of this foreign key.</td>
</tr>
</tbody>
</table>

The physical foreign key tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefColumns</td>
<td>References the physical columns (in the physical table that contains the foreign key) that match counterPartKeyRef. Each physical column is contained in a sub-element called either RefPhysicalColumn or RefCubeColumn.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<PhysicalForeignKey _
  hints=""
  counterPartKeyRef="">
  <Description />
  <RefColumns>
    <RefColumn columnRef="" .../>
    <RefColumn columnRef="" .../>
  </RefColumns>
</PhysicalForeignKey>
```

Example

```xml
<PhysicalForeignKey mdsid="m002377f9-0bbe-0000-714b-e31d00000000" name="CONTACTS_PRODUCTS_NOT_ORDERED_FKey1"
  counterPartKeyRef="/oracle/bi/server/base/PhysicalTable/Dim_W_PRODUCT_D_0022612d-0bb9-0000-714b-e31d00000000.xml#m00237f18-0bc0-0000-714b-e31d00000000">
  <RefColumns>
    <RefColumn refId="m002377f9-0bbe-0000-714b-e31d00000000-m0022ba69-0bbb-0000-714b-e31d00000000"
      columnRef="/oracle/bi/server/base/PhysicalTable/CONTACTS_PRODUCTS_NOT_00226232-0bb9-0000-714b-e31d00000000.xml#m0022ba69-0bbb-0000-714b-e31d00000000" />
  </RefColumns>
</PhysicalForeignKey>
```
3.10 MDS XML Elements: Complex Join

The complex join element corresponds to the complex join object in the Physical layer.

The complex join tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hints</td>
<td>For Oracle Databases only. Lists any database hints, such as index hints or leading hints, that were specified for this complex join.</td>
</tr>
<tr>
<td>cardinality</td>
<td>Indicates the cardinality of the join. Valid values are: ONE TO ONE, ZERO OR ONE TO ONE, ONE TO ZERO OR ONE, ZERO OR ONE TO ZERO OR ONE, ONE TO MANY, ZERO OR ONE TO MANY, ZERO OR ONE TO MANY, MANY TO ONE, MANY TO ZERO OR ONE, or MANY TO MANY.</td>
</tr>
<tr>
<td>extName</td>
<td>The external name for the join.</td>
</tr>
<tr>
<td>table1Ref</td>
<td>References the first physical table of the join.</td>
</tr>
<tr>
<td>table2Ref</td>
<td>References the second physical table of the join.</td>
</tr>
</tbody>
</table>

The complex join tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expr</td>
<td>Contains the expression that defines the relationship between the two tables.</td>
</tr>
<tr>
<td>ExternalExpr</td>
<td>Contains the external expression used for ADF Business Component data sources.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<ComplexJoin ...
  hints=""
  cardinality=""
  extName=""
  table1Ref=""
  table2Ref="">
  <Description />
  <Expr />
  <ExternalExpr />
</ComplexJoin>
```

Note: Both of the summary syntax elements `<Expr />` and `<ExternalExpr />` have the following sub-elements:

```xml
<ExprText>...</ExprText>
<ExprTextDesc>...</ExprTextDesc>
<ObjectRefList>
  <RefObject objectRef="/"/>
  <RefObject objectRef="/"/>
</ObjectRefList>
```

Example

```xml
<ComplexJoin mdsid="m00238388-0bc4-0000-714b-e31d00000000"
```
The cube table element corresponds to the physical cube table object in the Physical layer.

The cube table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The external name for the data source.</td>
</tr>
</tbody>
</table>
| type           | The type of physical cube table. Options are as follows:  
  - select indicates that this physical table object is a SELECT statement.  
  - sp indicates that this physical cube table is a stored procedure.  
  - alias indicates that this physical cube table is an alias table to a logical table source.  
  - none indicates that this physical cube table is a regular physical cube table. |
| isCacheable    | If set to true, indicates that this table is included in the Oracle BI Server query cache. |
### Table 3–19 (Cont.) Cube Table Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheExpiry</td>
<td>How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire.</td>
</tr>
<tr>
<td>isCacheEventTable</td>
<td>If set to true, indicates that this table has been registered as an Oracle BI Server event polling table.</td>
</tr>
<tr>
<td>pollFreq</td>
<td>The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds.</td>
</tr>
<tr>
<td>x</td>
<td>The x coordinate of the table when it is shown in the physical diagram.</td>
</tr>
<tr>
<td>y</td>
<td>The y coordinate of the table when it is shown in the physical diagram.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this physical cube table.</td>
</tr>
<tr>
<td>lastUpdated</td>
<td>The time at which row counts were last updated for this physical cube table.</td>
</tr>
<tr>
<td>scriptPath</td>
<td>For XML data sources only. The extended gateway script path.</td>
</tr>
<tr>
<td>isSysGen</td>
<td>If set to true, indicates that an internal utility has generated this physical cube table.</td>
</tr>
<tr>
<td>isOverrideCacheProp</td>
<td>For alias tables only. If set to true, indicates that the alias table has its own cache properties that override the cache properties of the source table.</td>
</tr>
<tr>
<td>hints</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>isDDLAggr</td>
<td>Indicates whether this cube table is an aggregate table.</td>
</tr>
<tr>
<td>memberReference</td>
<td>For Essbase data sources only. Indicates the cube property (for example, Member Name). An Essbase cube can have multiple aliases, and the value of memberReference is one of the alias names. The query engine will generate a query against the alias set for memberReference.</td>
</tr>
<tr>
<td>dateFormat</td>
<td>For Essbase only. Defines the format of date member, for example, 'yyyy-MM-dd.' The value is automatically set during import.</td>
</tr>
<tr>
<td>isMemberAliasUnique</td>
<td>This is used for query performance. If you know that all the member names in your data source are unique, you can set it to true by selecting the option Use unqualified member name for better performance.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the database for this cube table.</td>
</tr>
<tr>
<td>aliasTableVarRef</td>
<td>For Essbase data sources only. Each AliasTableVar represents an Essbase alias table.</td>
</tr>
</tbody>
</table>

The cube table tag includes the following child elements:

### Table 3–20 Cube Table Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>For Essbase data sources only. For any alias tables for this Essbase cube, lists the alias to use (such as default, LongNames, French, and so on).</td>
</tr>
<tr>
<td>DBMapItem</td>
<td>Stores the value of database-specific strings. The &lt;DBMapItem&gt; element contains a &lt;name&gt; attribute and a &lt;Value&gt; sub-element.</td>
</tr>
</tbody>
</table>
The syntax here includes the summary child element entry for the cube column element. The full syntax for the cube column element appears in Section 3.15, "MDS XML Elements: Cube Column."

```xml
<CubeTable
   extName=""
   type=""
   isCacheable=""
   cacheExpiry=""
   isCacheEventTable=""
   pollFreq=""
   x=""
   y=""
   rowCount=""
   lastUpdated=""
   scriptPath=""
```

### Table 3–20  (Cont.) Cube Table Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CubeVar</td>
<td>For SAP/BW data sources only. Cube variables are used as a means of parameterizing queries. Each cube variable is contained in a sub-element called CubeVar. This sub-element has the following attributes:</td>
</tr>
<tr>
<td></td>
<td>■ name: Name of the cube variable.</td>
</tr>
<tr>
<td></td>
<td>■ caption: A description (label or caption) associated with the cube variable, mainly used for display purposes.</td>
</tr>
<tr>
<td></td>
<td>■ type: The type of cube variable (SAP_VAR_TYPE_MEMBER, SAP_VAR_TYPE_HIERARCHY, or SAP_VAR_TYPE_NUMERIC).</td>
</tr>
<tr>
<td></td>
<td>■ selectionType: The selection type of the cube variable, for cube variables of type SAP_VAR_TYPE_MEMBER (SAP_VAR_SEL_TYPE_VALUE, SAP_SEL_TYPE_INTERVAL, or SAP_SEL_TYPE_COMPLEX).</td>
</tr>
<tr>
<td></td>
<td>■ entryType: Indicates whether replacing variables is optional or mandatory. Values include SAP_VAR_INPUT_TYPE_OPTIONAL, SAP_VAR_INPUT_TYPE_MANDATORY, or SAP_VAR_INPUT_TYPE_MANDATORY_NOT_INITIAL.</td>
</tr>
<tr>
<td></td>
<td>■ refDimension: The reference dimension. This column contains a DIMENSION_UNIQUE_NAME for the parameter type SAP_VAR_TYPE_HIERARCHY.</td>
</tr>
<tr>
<td></td>
<td>■ refHierarchy: The reference hierarchy. This column contains a HIERARCHY_UNIQUE_NAME for the variable type SAP_VAR_TYPE_MEMBER.</td>
</tr>
<tr>
<td></td>
<td>■ defaultLow: Contains a default value for the variable, or is zero.</td>
</tr>
<tr>
<td></td>
<td>■ defaultHigh: Contains a default value for the variable, or is zero. This property is only important for variables with the selection type SAP_VAR_SEL_TYPE_INTERVAL and SAP_VAR_SEL_TYPE_SELECTION.</td>
</tr>
<tr>
<td></td>
<td>■ overrideLow: The default value for the cube variable (used if defaultLow is zero).</td>
</tr>
<tr>
<td></td>
<td>■ overrideHigh: The default value for the cube variable (used if defaultHigh is zero).</td>
</tr>
<tr>
<td>CubeColumn</td>
<td>Specifies the cube column that belongs to this cube table. For details, see Section 3.15, &quot;MDS XML Elements: Cube Column.&quot;</td>
</tr>
</tbody>
</table>
3.12 MDS XML Elements: Physical Dimension

The physical dimension element corresponds to the physical dimension object in the Physical layer.

The physical dimension tag includes the following attributes:
The physical dimension tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefUserAttributes</td>
<td>References the dimension properties for this physical dimension. Each dimension property is contained in a sub-element called RefCubeColumn.</td>
</tr>
<tr>
<td>AlternateHierarchyName</td>
<td>For Essbase physical dimension objects, lists the names of any alternate hierarchies that belong to this dimension.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<PhysicalDimension _
  defaultMember=""
cubeTableRef=""
><Description />
<RefUserAttributes>
  <RefCubeColumn cubeColumnRef="" _/>
  <RefCubeColumn cubeColumnRef="" _/>
</RefUserAttributes>
<AlternateHierarchyName>
<![CDATA[ Calendar ]>]
</AlternateHierarchyName>
<AlternateHierarchyName>
<![CDATA[ Fiscal ]>]
</AlternateHierarchyName>
</PhysicalDimension>
```

Example

```xml
<PhysicalDimension mdsid="m80cb9c61-0bf8-0000-714b-e31d00000000"
  name="Months"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  cubeTableRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-714b-
  e31d00000000.xml#m80cb9c63-0bdf-0000-714b-e31d00000000">
  <RefUserAttributes>
    <RefCubeColumn refId="m80cb9c61-0bf8-0000-714b-e31d00000000-m80cb9c63-0bdf-0000-714b-
      e31d00000000" cubeColumnRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-
      714b-e31d00000000.xml#m80cb9c63-0bdf-0000-714b-e31d00000000" />
    <RefCubeColumn refId="m80cb9c61-0bf8-0000-714b-e31d00000000-m80cb9c6f-0be3-0000-714b-
      e31d00000000" cubeColumnRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-
      714b-e31d00000000.xml#m80cb9c6f-0be3-0000-714b-e31d00000000" />
  </RefUserAttributes>
  <AlternateHierarchyName>
    <![CDATA[ Calendar ]>]
  </AlternateHierarchyName>
  <AlternateHierarchyName>
    <![CDATA[ Fiscal ]>]
  </AlternateHierarchyName>
</PhysicalDimension>
```
3.13 MDS XML Elements: Physical Hierarchy

The physical hierarchy element corresponds to the physical hierarchy object in the Physical layer.

The physical hierarchy tag includes the following attributes:

Table 3–23 Physical Hierarchy Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memberType</td>
<td>Values are Unknown, Regular, All, Formula, Measure. Default value is Unknown.</td>
</tr>
<tr>
<td>extName</td>
<td>The external name for the physical hierarchy.</td>
</tr>
<tr>
<td>type</td>
<td>The type of physical hierarchy. Options include FULLY BALANCED, UNBALANCED, RAGGED BALANCED, and NETWORK. This option corresponds to the Hierarchy Type option in the Physical Hierarchy dialog in the Administration Tool.</td>
</tr>
<tr>
<td>dimType</td>
<td>The dimension type (for example, Measure Dimension) for the dimension to which this hierarchy belongs.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this physical hierarchy.</td>
</tr>
<tr>
<td>isTimeDim</td>
<td>If set to true, indicates that this physical dimension is a time dimension.</td>
</tr>
<tr>
<td>dimUniqueName</td>
<td>The name of the dimension to which this hierarchy belongs.</td>
</tr>
<tr>
<td>isMemberAliasUnique</td>
<td>Indicates whether member names (including aliases) are unique in a given hierarchy so that the Oracle BI Server can take advantage of specific MDX syntax to optimize performance. This option corresponds to the Use unqualified member name for better performance option in the Physical Hierarchy dialog in the Administration Tool.</td>
</tr>
<tr>
<td>cubeTableRef</td>
<td>References the cube table for this physical hierarchy.</td>
</tr>
<tr>
<td>dimensionRef</td>
<td>References the physical dimension for this physical hierarchy.</td>
</tr>
<tr>
<td>memberKeyRef</td>
<td>For parent-child (value) hierarchies. References the cube column that is the member key for this hierarchy.</td>
</tr>
<tr>
<td>parentKeyRef</td>
<td>For parent-child (value) hierarchies. References the cube column that is the parent key for this hierarchy.</td>
</tr>
</tbody>
</table>

The physical hierarchy tag includes the following child elements:

Table 3–24 Physical Hierarchy Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefColumns</td>
<td>References the set of cube columns that belong to this physical hierarchy. Each cube column is contained in a sub-element called RefCubeColumn.</td>
</tr>
<tr>
<td>PhysicalLevel</td>
<td>For level-based hierarchies. References the set of physical levels that belong to this physical hierarchy. Each level is contained in a sub-element called RefPhysicalLevel. For more details, see Section 3.14, &quot;MDS XML Elements: Physical Level.&quot;</td>
</tr>
</tbody>
</table>
Syntax

The syntax here includes the summary child element entry for the physical level element. The full syntax for the physical level element appears in Section 3.14, "MDS XML Elements: Physical Level."

```
<PhysicalHierarchy _
    memberType="" extName=""
    type="" dimType=""
    rowCount=""
    isTimeDim=""
    dimUniqueName=""
    isMemberAliasUnique=""
    cubeTableRef=""
    dimensionRef=""
    memberKeyRef=""
    parentKeyRef="">
    <Description />
    <RefColumns>
        <RefCubeColumn cubeColumnRef="" _/>
        <RefCubeColumn cubeColumnRef="" _/>
    </RefColumns>
</PhysicalHierarchy>
```

Example

```
<PhysicalHierarchy mdsid="m80cb9c58-0be1-0000-714b-e31d00000000"
    name="Employees"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    memberType="all"
    extName="Employees"
    type="VALUE"
    dimType="Other"
    dimUniqueName="Employees"
    cubeTableRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-714b-
        e31d00000000.xml#m80cb9c63-0bdf-0000-714b-e31d00000000"
    dimensionRef="/oracle/bi/server/base/PhysicalDimension/Employees_80cb9c5d-0bf8-
        0000-714b-e31d00000000.xml#m80cb9c5d-0bf8-0000-714b-e31d00000000"
    memberKeyRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-714b-
        e31d00000000.xml#m80cb9c9c-0be3-0000-714b-e31d00000000"
    parentKeyRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-714b-
        e31d00000000.xml#m80cb9cda-0be3-0000-714b-e31d00000000">
    <RefColumns>
        <RefCubeColumn refId="m80cb9c58-0be1-0000-714b-e31d00000000-m80cb9c9c-0be3-
            0000-714b-e31d00000000"
            cubeColumnRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-
                714b-e31d00000000.xml#m80cb9c9c-0be3-0000-714b-e31d00000000" />
        <RefCubeColumn refId="m80cb9c58-0be1-0000-714b-e31d00000000-m80cb9c9c-0be3-
            0000-714b-e31d00000000"
            cubeColumnRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-
                714b-e31d00000000.xml#m80cb9c9c-0be3-0000-714b-e31d00000000" />
        <RefCubeColumn refId="m80cb9c58-0be1-0000-714b-e31d00000000-m80cb9c9c-0be3-
            0000-714b-e31d00000000"
            cubeColumnRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-
                714b-e31d00000000.xml#m80cb9c9c-0be3-0000-714b-e31d00000000" />
        <RefCubeColumn refId="m80cb9c58-0be1-0000-714b-e31d00000000-m80cb9c9c-0be3-
            0000-714b-e31d00000000"
            cubeColumnRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-
                714b-e31d00000000.xml#m80cb9c9c-0be3-0000-714b-e31d00000000" />
        <RefCubeColumn refId="m80cb9c58-0be1-0000-714b-e31d00000000-m80cb9c9c-0be3-
            0000-714b-e31d00000000"
            cubeColumnRef="/oracle/bi/server/base/CubeTable/Sample_80cb9c63-0bdf-0000-
                714b-e31d00000000.xml#m80cb9c9c-0be3-0000-714b-e31d00000000" />
```

3.14 MDS XML Elements: Physical Level

The physical level element corresponds to the hierarchical levels for a physical cube table object in the Physical layer.

The physical level element is not a top-level element - it appears as a child element of the physical hierarchy element.

The physical level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>levelNum</td>
<td>Identifies the order of this physical level in the hierarchy.</td>
</tr>
<tr>
<td>extName</td>
<td>Fully qualified name for the level.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this physical level.</td>
</tr>
<tr>
<td>levelKeyRef</td>
<td>References the key column for this physical level.</td>
</tr>
</tbody>
</table>
Syntax

```xml
<PhysicalLevel _
   levelNumber="" 
   extName="" 
   rowCount="" 
   levelKeyRef=""> 
   <Description />
</PhysicalLevel>
```

Example

```xml
<PhysicalLevel mdsid="m80cb9d03-0be5-0000-714b-e31d00000000"
   name='All Employees'
   levelNum='0'
   extName='All Employees'
   levelKeyRef="/oracle/bi/server/base/CubeTable/Sample80cb9c63-0bdf-0000-714b-
   e31d00000000.xml#m80cb9c8d-0be3-0000-714b-e31d00000000">
</PhysicalLevel>
```

3.15 MDS XML Elements: Cube Column

The cube column element corresponds to the cube column of a physical cube object in the Physical layer.

The cube column element is not a top-level element - it appears as a child element of the cube table element.

The cube column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The external name for the cube column. This attribute is required if the same name (such as STATE) is used in multiple hierarchies.</td>
</tr>
<tr>
<td>aggrType</td>
<td>The type of aggregation on the column, if any.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the cube column, such as VARCHAR.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the column. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the cube column.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the column.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this cube column.</td>
</tr>
<tr>
<td>lastUpdated</td>
<td>The time at which row counts were last updated for this cube column.</td>
</tr>
</tbody>
</table>
### MDS XML Elements: Cube Column

**Syntax**

```xml
<CubeColumn _
    extName=""
    aggrType=""
    dataType=""
    nullable=""
    precision=""
    rowCount=""
    lastUpdated=""
    scale="">
    dimPropType=""
    attributeHierarchyRef=""
    physicalLevelRef=""
</CubeColumn>
```

**Example**

```xml
<CubeColumn mdsid="m80cbf77f-0be3-0000-714b-e31d00000000"
    name="Product Types - Ancestor Ref"
    extName="Ancestor_Names"
    dataType="VARCHAR"
    precision="128"
    dimPropType="ancestorRef"
    physicalLevelRef="/oracle/bi/server/base/PhysicalHierarchy/Products_80cbf730-0be1-0000-714b-e31d00000000.xml#m80cbf7ef-0be5-0000-714b-e31d00000000">
    <Description />
</CubeColumn>
```

### Table 3–26 (Cont.) Cube Column Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimPropType</td>
<td>For Essbase dimensional columns only. Indicates the property type of the cube column. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- <strong>uda</strong>: User Defined Attribute (UDA) type.</td>
</tr>
<tr>
<td></td>
<td>- <strong>memnor</strong>: Memnor type, used for outline sorts in the logical layer. Imported at the lowest level of each dimension. Called <strong>Outline Sort</strong> in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>- <strong>attribute</strong>: Attribute type, for attribute dimensions.</td>
</tr>
<tr>
<td></td>
<td>- <strong>other</strong>: The type is different than those listed, or unknown.</td>
</tr>
<tr>
<td></td>
<td>- <strong>none</strong>: The column does not have a property type. Note that this type is mapped to <strong>Member Alias</strong> in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>- <strong>ancestorRef</strong>: References the ancestor of a dimension.</td>
</tr>
<tr>
<td></td>
<td>- <strong>memberKey</strong>: The column is a member key.</td>
</tr>
<tr>
<td></td>
<td>- <strong>levelNumber</strong>: The column is the lowest member of the hierarchy. Called <strong>Leaf</strong> in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>- <strong>genNumber</strong>: The column is the root member of the hierarchy. Called <strong>Root</strong> in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>- <strong>parentRef</strong>: References the parent of a dimension.</td>
</tr>
<tr>
<td>attributeHierarchyRef</td>
<td>References the physical hierarchy to which this cube column belongs.</td>
</tr>
<tr>
<td>physicalLevelRef</td>
<td>References the physical level to which this cube column belongs.</td>
</tr>
</tbody>
</table>
3.16 MDS XML Elements: Analytic Workspace

The Analytic Workspace element corresponds to the Oracle OLAP Analytic Workspace (AW) object in the Physical layer.

The Analytic Workspace tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of the Analytic Workspace.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the physical catalog for this Analytic Workspace.</td>
</tr>
<tr>
<td>dynamicNameRef</td>
<td>If a session variable is being used to specify the name of this Analytic Workspace, this element references that session variable.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<AnalyticWorkspace version="" containerRef="" dynamicNameRef="">
  <Description />
</AnalyticWorkspace>
```

Example

```xml
<AnalyticWorkspace mdsid="m000000ae-0be9-0000-714b-e31d00000000" name="GLOBAL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.oracle.com/obis/repository" version="10" containerRef="/oracle/bi/server/base/PhysicalCatalog/GLOBAL_000000ad-0bd1-0000-714b-e31d00000000.xml#m000000ad-0bd1-0000-714b-e31d00000000">
  <Description />
</AnalyticWorkspace>
```

3.17 MDS XML Elements: AW Cube

The AW cube element corresponds to the Oracle OLAP cube object in the Physical layer.

The AW cube tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the cube in physical SQL queries.</td>
</tr>
<tr>
<td>isCacheable</td>
<td>If set to true, indicates that this table is included in the Oracle BI Server query cache.</td>
</tr>
<tr>
<td>CacheExpiry</td>
<td>How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire.</td>
</tr>
<tr>
<td>isCacheEventTable</td>
<td>If set to true, indicates that this table has been registered as an Oracle BI Server event polling table.</td>
</tr>
<tr>
<td>pollFreq</td>
<td>The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds.</td>
</tr>
</tbody>
</table>
The AW cube tag includes the following child elements:

**Table 3–29   AW Cube Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompositeInfo</td>
<td>References dimensions for the cube. Contains the sub-element called Composite, which has one attribute called name and a sub-element called Dimension. Each dimension is contained in the Dimension sub-element.</td>
</tr>
<tr>
<td>AWColumn</td>
<td>References the Oracle OLAP columns that belong to this Oracle OLAP cube. For more details, see Section 3.21, &quot;MDS XML Elements: AW Column.&quot;</td>
</tr>
</tbody>
</table>

**Syntax**

The syntax here includes the summary child element entry for the AW column element. The full syntax for the AW column element appears in Section 3.21, "MDS XML Elements: AW Cube."
XML Elements: AW Column.

```xml
<AWCube _
   extName=""
   isCacheable=""
   cacheExpiry=""
   isCacheEventTable=""
   pollFreq=""
   x=""
   y=""
   rowCount=""
   lastUpdated=""
   hints=""
   isExtGatewayInProcess=""
   isOverrideCacheProp=""
   isDDLAggr=""
   isSysGen=""
   isPartitioned=""
   densityStatus=""
   materializationStatus=""
   aggrConnPoolRef=""
   containerRef=""
   <Description />
   <CompositeInfo>
     <Composite name="">
       <Dimension _/>
     </Composite>
   </CompositeInfo>
   <AWColumn mdsid="m000000d4-0bf4-0000-714b-e31d00000000"
     name="Units"
     extName="TIME_CHANNEL_CUBE_UNITS"
     dataType="DOUBLE" nullable="true">
   </AWColumn>
   </AWCube>

Example

<AWCube mdsid="m000000b0-0bee-0000-714b-e31d00000000"
   name="Time Channel Cube"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.oracle.com/obis/repository"
   extName="Time Channel Cube"
   isPartitioned="true"
   containerRef="/oracle/bi/server/base/AnalyticWorkspace/GLOBAL_000000ae-0be9-0000-714b-e31d00000000.xml#m000000ae-0be9-0000-714b-e31d00000000">
   <CompositeInfo>
     <Composite name="TIME_CHANNEL_CUBE_102_PRTCOMP">
       <Dimension>
         <![CDATA[ CHANNEL ]]> 
       </Dimension>
     </Composite>
     <Composite name="TIME_CHANNEL_CUBE_119_PRTCOMP">
       <Dimension>
         <![CDATA[ CHANNEL ]]> 
       </Dimension>
     </Composite>
     ...
     ...
     <Composite name="TIME_CHANNEL_CUBE_UNITS">
       <Dimension>
         <![CDATA[ CHANNEL ]]> 
       </Dimension>
     </Composite>
   </CompositeInfo>
   <AWColumn mdsid="m000000d4-0bf4-0000-714b-e31d00000000"
     name="Units"
     extName="TIME_CHANNEL_CUBE_UNITS"
     dataType="DOUBLE" nullable="true">
   </AWColumn>
   <AWColumn mdsid="m000000d5-0bf4-0000-714b-e31d00000000"
3.18 MDS XML Elements: AW Dimension

The AW dimension element corresponds to the Oracle OLAP dimension object in the Physical layer.

The AW dimension tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The dimension type.</td>
</tr>
<tr>
<td>isCacheable</td>
<td>If set to true, indicates that this dimension is included in the Oracle BI Server query cache.</td>
</tr>
<tr>
<td>CacheExpiry</td>
<td>If set to true, indicates that this dimension is included in the Oracle BI Server query cache.</td>
</tr>
<tr>
<td>isCacheEventTable</td>
<td>How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire.</td>
</tr>
<tr>
<td>isOverrideCacheProp</td>
<td>For alias tables only. If set to true, indicates that the alias table has its own cache properties that override the cache properties of the source table.</td>
</tr>
<tr>
<td>pollFreq</td>
<td>If set to true, indicates that this table has been registered as an Oracle BI Server event polling table.</td>
</tr>
<tr>
<td>x</td>
<td>The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds.</td>
</tr>
<tr>
<td>y</td>
<td>The x coordinate of the table when it is shown in the physical diagram.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The y coordinate of the table when it is shown in the physical diagram.</td>
</tr>
<tr>
<td>lastUpdated</td>
<td>The time at which row counts were last updated for this Oracle OLAP dimension.</td>
</tr>
<tr>
<td>hints</td>
<td>This attribute is not used.</td>
</tr>
</tbody>
</table>
The AW dimension tag includes the following child elements:

**Table 3–31 AW Dimension Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWObject</td>
<td>References the Oracle OLAP objects, such a list of hierarchies, associated with this dimension. The AWObject element includes the following attributes: <img src="https://via.placeholder.com/150.png" alt="image" /></td>
</tr>
<tr>
<td>AWColumn</td>
<td>References the AW columns that belong to this AW dimension. For more details, see Section 3.21, &quot;MDS XML Elements: AW Column.&quot;</td>
</tr>
</tbody>
</table>

**Syntax**

The syntax here includes the summary child element entry for the AW column element. The full syntax for the AW column element appears in Section 3.21, "MDS XML Elements: AW Column."

```
<AWDimension _
  type=""
  isCacheable=""
  cacheExpiry=""
  isCacheEventTable=""
  isOverrideCacheProp=""
  pollFreq=""
```
MDS XML Elements: AW Dimension

Example

```xml
<AWDimension mdsid="m000000b2-0bec-0000-714b-e31d00000000"
name="Channel" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.oracle.com/obis/repository"
type="none"
extName="CHANNEL"
containerRef="/oracle/bi/server/base/AnalyticWorkspace/GLOBAL_000000ae-0be9-
0000-714b-e31d00000000.xml#m000000ae-0be9-0000-714b-e31d00000000">
  <AWObject type="HIER_LIST" name="CHANNEL_HIERLIST"
dataType="VARCHAR" precision="100" />
  <AWObject type="LEVEL_LIST" name="CHANNEL_LEVELLIST"
dataType="VARCHAR" precision="100" />
  <AWObject type="IN_HIER" name="CHANNEL_INHIER"
dataType="VARCHAR" precision="100" />
  <AWObject type="PARENT_REL" name="CHANNEL_PARENTREL"
dataType="VARCHAR" precision="100" />
  <AWObject type="LEVEL_REL" name="CHANNEL_LEVELREL"
dataType="VARCHAR" precision="100" />
  <AWObject type="FAMILY_REL" name="CHANNEL_FAMILYREL"
dataType="VARCHAR" precision="100" />
  <AWColumn mdsid="m000000fc-0bf4-0000-714b-e31d00000000"
name="Total Channel - Long Description"
extName="CHANNEL_LONG_DESCRIPTION"
awLevelRef="/oracle/bi/server/base/AWHierarchy/Primary_0000012f-0bf0-0000-
714b-e31d00000000.xml#m0000013b-0bf2-0000-714b-e31d00000000"
 />
  <AWColumn mdsid="m000000fd-0bf4-0000-714b-e31d00000000"
name="Total Channel - Short Description"
extName="CHANNEL_SHORT_DESCRIPTION"
awLevelRef="/oracle/bi/server/base/AWHierarchy/Primary_0000012f-0bf0-0000-
714b-e31d00000000.xml#m0000013b-0bf2-0000-714b-e31d00000000"
 />
  <AWColumn mdsid="m000000fe-0bf4-0000-714b-e31d00000000"
name="Total Channel"
extName="CHANNEL"
awLevelRef="/oracle/bi/server/base/AWHierarchy/Primary_0000012f-0bf0-0000-
714b-e31d00000000.xml#m0000013b-0bf2-0000-714b-e31d00000000"
 /></AWDimension>
```
3.19 MDS XML Elements: AW Hierarchy

The AW hierarchy element corresponds to the Oracle OLAP hierarchy object in the Physical layer.

The AW hierarchy tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of Oracle OLAP hierarchy (value or level).</td>
</tr>
<tr>
<td>isDefault</td>
<td>If set to true, indicates that this is the default hierarchy for the Oracle OLAP dimension.</td>
</tr>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the hierarchy in physical SQL queries.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the Oracle OLAP hierarchy, such as VARCHAR.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the Oracle OLAP hierarchy. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the Oracle OLAP hierarchy.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the Oracle OLAP hierarchy.</td>
</tr>
<tr>
<td>memberKeyRef</td>
<td>References the member key (for parent-child hierarchies).</td>
</tr>
<tr>
<td>dimensionRef</td>
<td>References the AW dimension for this Oracle OLAP hierarchy.</td>
</tr>
</tbody>
</table>

The AW hierarchy tag includes the following child elements:
### Syntax

The syntax here includes the summary child element entry for the AW level element. The full syntax for the AW level element appears in Section 3.20, "MDS XML Elements: AW Level."

```xml
<AWHierarchy ...
    type="" isDefault="" extName="" dataType="" nullable="" precision="" scale="" memberKeyRef="" dimensionRef="">
    <Description />
    <AWLevel />
</AWHierarchy>
```

### Example

```xml
<AWHierarchy mdsid="m0000012d-0bf0-0000-714b-e31d00000000"
    name="Calendar Year"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    isDefault="true"
    extName="CALENDAR_YEAR"
    dataType="UNKNOWN"
    dimensionRef="/oracle/bi/server/base/AWDimension/Time_000000b1-0bec-0000-714b-e31d00000000.xml#m000000b1-0bec-0000-714b-e31d00000000">
    <AWLevel mdsid="m00000133-0bf2-0000-714b-e31d00000000"
        name="All Years"
        extName="ALL_YEARS"
        dataType="UNKNOWN"
        levelKeyRef="/oracle/bi/server/base/AWDimension/Time_000000b1-0bec-0000-714b-e31d00000000.xml#m000000da-0bf4-0000-714b-e31d00000000"
        awHierarchyRef="/oracle/bi/server/base/AWHierarchy/Calendar Year_0000012d-0bf0-0000-714b-e31d00000000.xml#m0000012d-0bf0-0000-714b-e31d00000000"/>
</AWLevel>
<AWLevel mdsid="m00000134-0bf2-0000-714b-e31d00000000"
    name="Year"
    extName="YEAR"
    dataType="UNKNOWN"
    levelKeyRef="/oracle/bi/server/base/AWDimension/Time_000000b1-0bec-0000-714b-e31d00000000.xml#m000000da-0bf4-0000-714b-e31d00000000"
    awHierarchyRef="/oracle/bi/server/base/AWHierarchy/Calendar Year_0000012d-0bf0-0000-714b-e31d00000000.xml#m0000012d-0bf0-0000-714b-e31d00000000"/>
</AWLevel>
<AWLevel mdsid="m00000135-0bf2-0000-714b-e31d00000000"
    name="Quarter"
    extName="QUARTER"
</AWLevel>
```

### Table 3–33 AW Hierarchy Child Elements

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWLevel</td>
<td>For level-based hierarchies only. References the AW level for this Oracle OLAP dimension. For more details, see Section 3.20, &quot;MDS XML Elements: AW Level.&quot;</td>
</tr>
</tbody>
</table>
3.20 MDS XML Elements: AW Level

The AW level element corresponds to the Oracle OLAP level object in the Physical layer.

The AW level element is not a top-level element - it appears as a child element of the AW hierarchy element.

The AW level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the level in physical SQL queries.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the Oracle OLAP level, such as VARCHAR.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the Oracle OLAP level. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the Oracle OLAP level.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the Oracle OLAP level.</td>
</tr>
<tr>
<td>levelKeyRef</td>
<td>References the key column for this Oracle OLAP level.</td>
</tr>
<tr>
<td>awHierarchyRef</td>
<td>References the AW hierarchy for this Oracle OLAP level.</td>
</tr>
</tbody>
</table>

The AW level tag does not include any child elements.

Syntax

```xml
<AWLevel
   extName=''
dataType=''
nullable=''
precision=''
scale=''
levelKeyRef=''
awHierarchyRef=''
><Description />
</AWLevel>
```
Example

```xml
<AWLevel mdsid="m00000133-0bf2-0000-714b-e31d00000000"
    name="All Years"
    extName="ALL_YEARS"
    dataType="UNKNOWN"
    levelKeyRef="/oracle/bi/server/base/AWDimension/Time_000000b1-0bec-0000-714b-
    e31d00000000.xml#m000000da-0bf4-0000-714b-e31d00000000"
    awHierarchyRef="/oracle/bi/server/base/AWHierarchy/Calendar Year_0000012d-0bf0-0000-714b-e31d00000000.xml#m0000012d-0bf0-0000-714b-e31d00000000">
</AWLevel>
```

3.21 MDS XML Elements: AW Column

The AW column element corresponds to the AW column object in the Physical layer. The AW column element is not a top-level element - it appears as a child element of the AW cube and AW dimension elements.

The AW column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the column in physical SQL queries.</td>
</tr>
<tr>
<td>isParentKey</td>
<td>Indicates whether this Oracle OLAP column is a parent key for a level.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the Oracle OLAP column, such as VARCHAR.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the Oracle OLAP column.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the Oracle OLAP column. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the Oracle OLAP column.</td>
</tr>
<tr>
<td>awHierarchyRef</td>
<td>For value-based hierarchies, references the AW hierarchy for this Oracle OLAP column.</td>
</tr>
<tr>
<td>awLevelRef</td>
<td>References the hierarchy level for this Oracle OLAP column.</td>
</tr>
</tbody>
</table>

The AW column tag does not include any child elements.

Syntax

```xml
<AWColumn _
    extName=""
    isParentKey=""
    dataType=""
    nullable=""
    precision=""
    scale=""
    awHierarchyRef=""
    awLevelRef="/>
    <Description />
</AWColumn>
```

Example

```xml
<AWColumn mdsid="m000000fe-0bf4-0000-714b-e31d00000000"
```
name='Total Channel'
extName='CHANNEL'
dataType='VARCHAR' precision='100'
awLevelRef='/oracle/bi/server/base/AWHierarchy/Primary_0000012f-0bf0-0000-714b-e31d00000000.xml#m0000013b-0bf2-0000-714b-e31d00000000'">
</AWColumn>
This chapter provides MDS XML reference information for presentation elements. Presentation elements typically correspond to objects in the Presentation layer of the Oracle BI repository, or in other words, the objects that appear in the left-hand pane of the Oracle BI Administration Tool.

This chapter contains the following topics:

- Section 4.1, "MDS XML Elements: Presentation Catalog"
- Section 4.2, "MDS XML Elements: Presentation Table"
- Section 4.3, "MDS XML Elements: Presentation Column"
- Section 4.4, "MDS XML Elements: Presentation Hierarchy"
- Section 4.5, "MDS XML Elements: Presentation Level"

### 4.1 MDS XML Elements: Presentation Catalog

The presentation catalog element corresponds to the subject area object in the Presentation layer.

The presentation catalog tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isExportKeys</td>
<td>If set to true, indicates that logical keys for this subject area are exposed to other applications.</td>
</tr>
<tr>
<td>isAutoAggr</td>
<td>If set to true, indicates that auto aggregation is enabled. Auto aggregation is the process by which the Oracle BI Server infers the GROUP BY from the attributes projected in the query. Typically, auto aggregation is always enabled for queries against the Presentation layer.</td>
</tr>
<tr>
<td>hasDispName</td>
<td>Indicates whether this subject area has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this subject area.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this subject area has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this subject area.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the container for this presentation catalog.</td>
</tr>
<tr>
<td>subjectAreaRef</td>
<td>References the business model for this subject area.</td>
</tr>
</tbody>
</table>
The presentation catalog tag includes the following child elements:

### Table 4–2 Presentation Catalog Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Lists any aliases that have been defined for this subject area. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias has an attribute called <code>name</code>.</td>
</tr>
<tr>
<td>RefTables</td>
<td>References the presentation tables that belong to this subject area. Each presentation table is contained in a sub-element called <code>RefPresentationTable</code>.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the <code>Hide object if</code> field in the Administration Tool. The expression is contained in two sub-elements called <code>ExprText</code> and <code>ExprTextDesc</code>.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<PresentationCatalog _
  isExportKeys=""
  isAutoAggr=""
  hasDispName=""
  dispName=""
  hasDispDescription=""
  dispDescription=""
  containerRef=""
  subjectAreaRef=""
  defaultFactColumnRef=""
  <Description />
  <Alias name=""/>
  <RefTables>
    <RefPresentationTable presentationTableRef="" _/>
    <RefPresentationTable presentationTableRef="" _/>
  </RefTables>
  <VisibilityFilter>
    <ExprText _/>
    <ExprTextDesc _/>
  </VisibilityFilter>
</PresentationCatalog>
```

Example

```xml
<PresentationCatalog mdsid="m80d0301d-0fa4-0000-714b-e31d00000000"
  name="Inventory Org Real Time"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  hasDispName="false"
```

The presentation catalog tag includes the following child elements:

### Table 4–1 (Cont.) Presentation Catalog Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>defaultFactColRef</td>
<td>References the implicit fact column for this subject area, if one has been set. This column is used to specify a default join path between dimension tables when there are several possible alternatives or contexts.</td>
</tr>
</tbody>
</table>
4.2 MDS XML Elements: Presentation Table

The presentation table element corresponds to the presentation table object in the Presentation layer.

The presentation table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation table has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this presentation table.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this presentation table has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this presentation table.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the presentation catalog for this presentation table.</td>
</tr>
</tbody>
</table>

The presentation table tag includes the following child elements:
Table 4–4 Presentation Table Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Lists any aliases that have been defined for this presentation table. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias has an attribute called name.</td>
</tr>
<tr>
<td>PresentationColumn</td>
<td>References the presentation columns that belong to this presentation table. For details, see Section 4.3, &quot;MDS XML Elements: Presentation Column.&quot;</td>
</tr>
<tr>
<td>RefHierarchies</td>
<td>References the hierarchies for this presentation table. Each hierarchy is contained in a sub-element called RefPresentationHierarchy.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in two sub-elements called ExprText and ExprTextDesc.</td>
</tr>
<tr>
<td>RefContentFolders</td>
<td>References any child presentation tables that have been designated for this presentation table. Child presentation tables appear as nested folders in Answers and BI Composer. Note that the folders only appear to be nested - they are not actually nested in terms of drill-down, and the qualified names of the objects remain the same. Each child presentation table is contained in a sub-element called RefPresentationTable.</td>
</tr>
</tbody>
</table>

Syntax

The syntax here includes the summary child element entry for the presentation column element. The full syntax for the presentation column element appears in Section 4.3, "MDS XML Elements: Presentation Column."

```xml
<PresentationTable
  hasDispName=""
  dispName=""
  hasDispDescription=""
  dispDescription=""
  containerRef="">
  <Description />
  <Alias name="" />
  <PresentationColumn />
  <RefHierarchies>
    <RefPresentationHierarchy presentationHierarchyRef="" />}
    <RefPresentationHierarchy presentationHierarchyRef="" />}
  </RefHierarchies>
  <VisibilityFilter>
    <ExprText />
    <ExprTextDesc />
  </VisibilityFilter>
  <RefContentFolders>
    <RefPresentationTable presentationTableRef="" />
  </RefContentFolders>
</PresentationTable>
```
Example

```xml
<PresentationTable mdsid="m80ebeef8-0fa8-0000-714b-e31d00000000"
    name="Class"
    iconIndex="93"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.oracle.com/obis/repository"
hasDispName="false"
hasDispDescription="false"
containerRef="/oracle/bi/server/base/PresentationCatalog/OTBI HCM Stock
Details_80ebeef7-0fa4-0000-714b-e31d00000000.xml#m80ebeef7-0fa4-0000-714b-
e31d00000000">
    <Alias name="Dim - Class" />
    <PresentationColumn mdsid="m80ebefb7-0faa-0000-714b-e31d00000000"
        name="Class"
        hasDispName="false"
        hasDispDescription="false"
        overrideLogicalName="false"
        logicalColumnRef="/oracle/bi/server/base/LogicalTable/Dim _u0150 Class_
80dc7e5c-07f3-0000-714b-e31d00000000.xml#m80dbc57a-07d6-0000-714b-
e31d00000000"
    />
    <PresentationColumn mdsid="m80ebefb9-0faa-0000-714b-e31d00000000"
        name="ClassType"
        hasDispName="false"
        hasDispDescription="false"
        overrideLogicalName="false"
        logicalColumnRef="/oracle/bi/server/base/LogicalTable/Dim _u0150 Class_
80dc7e5c-07f3-0000-714b-e31d00000000.xml#m80dbc580-07d6-0000-714b-
e31d00000000"
    />
</PresentationTable>
</PresentationColumn>
<RefHierarchies>
    <RefPresentationHierarchy refId="m80ebeef8-0fa8-0000-714b-e31d00000000-
m80ebefb7-0faa-0000-714b-e31d00000000"
        presentationHierarchyRef="/oracle/bi/server/base/PresentationHierarchy/
        Class_80ebeef9-0fbc-0000-714b-e31d00000000.xml#m80ebeef9-0fbc-0000-714b-
e31d00000000" />
</RefHierarchies>
</PresentationTable>
```

4.3 MDS XML Elements: Presentation Column

The presentation column element corresponds to the presentation column object in the Presentation layer.

The presentation column element is not a top-level element - it appears as a child element of the presentation table element.

The presentation column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>overrideLogicalName</td>
<td>Indicates whether the presentation column has been configured to always use the logical column name. This option corresponds to Use Logical Column Name in the Administration Tool. A false value for this attribute indicates that this option has been selected.</td>
</tr>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation column has a custom display name.</td>
</tr>
</tbody>
</table>
The presentation column tag includes the following child elements:

**Table 4–6 Presentation Column Child Elements**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Lists any aliases that have been defined for this presentation column. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias has an attribute called name.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in two sub-elements called ExprText and ExprTextDesc.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<PresentationColumn _
    overrideLogicalName=""
    hasDispName=""
    dispName=""
    hasDispDescription=""
    dispDescription=""
    logicalColumnRef=""/>
  <Description />
  <Alias name="" />
  <VisibilityFilter>
    <ExprText _/>
    <ExprTextDesc _/>
  </VisibilityFilter>
</PresentationColumn>
```

**Example**

```xml
<PresentationColumn mdsid='m80ebefb7-0faa-0000-714b-e31d00000000'
    name="Class"
    hasDispName="false"
    hasDispDescription="false"
    overrideLogicalName="false"
    logicalColumnRef="/oracle/bi/server/base/LogicalTable/Dim_u0150 Class_80dc7e5c-07f3-0000-714b-e31d00000000.xml#m80dbc57a-07d6-0000-714b-e31d00000000">
</PresentationColumn>
```
4.4 MDS XML Elements: Presentation Hierarchy

The presentation hierarchy element corresponds to the presentation hierarchy object in the Presentation layer.

The presentation hierarchy tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation hierarchy has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this presentation hierarchy.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this presentation hierarchy has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this presentation hierarchy.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the presentation table for this presentation hierarchy.</td>
</tr>
<tr>
<td>logicalDimensionRef</td>
<td>References the logical dimension upon which this presentation hierarchy is based.</td>
</tr>
</tbody>
</table>

The presentation hierarchy tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PresentationLevel</td>
<td>References the presentation levels for this presentation hierarchy. For details, see Section 4.5, &quot;MDS XML Elements: Presentation Level.&quot;</td>
</tr>
<tr>
<td>Alias</td>
<td>Lists any aliases that have been defined for this presentation hierarchy. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias has an attribute called name.</td>
</tr>
<tr>
<td>RefDisplayColumns</td>
<td>For parent-child hierarchies only. This element references the columns designated to be used for display for this parent-child hierarchy. Each display column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in two sub-elements called ExprText and ExprTextDesc.</td>
</tr>
</tbody>
</table>

Syntax

The syntax here includes the summary child element entry for the presentation level element. The full syntax for the presentation level element appears in Section 4.5, "MDS XML Elements: Presentation Level."

```xml
<PresentationHierarchy _
    hasDisplayName=""
    displayName=""
    hasDispDescription=""
```
Example

```xml
<PresentationHierarchy mdsid="m80bee414-0fbc-0000-714b-e31d00000000" name="Product - Category" xmlns="http://www.oracle.com/obis/repository" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" hasDispName="false" hasDispDescription="false" containerRef="/oracle/bi/server/base/PresentationHierarchy/80bee414-0fbc-0000-714b-e31d00000000.xml#m80bee414-0fbc-0000-714b-e31d00000000" logicalLevelRef="/oracle/bi/server/base/Dimension/80beaf01-07e3-0000-714b-e31d00000000.xml#m80beaf01-07e3-0000-714b-e31d00000000">
  <RefDisplayColumns>
    <RefPresentationColumn refId="m80bee419-0fbe-0000-714b-e31d00000000" presentationColumnRef="/oracle/bi/server/base/PresentationTable/80beb309-0fa8-0000-714b-e31d00000000.xml#m80beb309-0fa8-0000-714b-e31d00000000"/>
    <RefPresentationColumn refId="m80bee41a-0fbe-0000-714b-e31d00000000" presentationColumnRef="/oracle/bi/server/base/PresentationTable/80beb309-0fa8-0000-714b-e31d00000000.xml#m80beb2a9-0fa8-0000-714b-e31d00000000"/>
  </RefDisplayColumns>
</PresentationHierarchy>
```

4.5 MDS XML Elements: Presentation Level

The presentation level element corresponds to the presentation level object in the Presentation layer.

The presentation level element is not a top-level element - it appears as a child element of the presentation hierarchy element.

The presentation level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation level has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this presentation level.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this presentation level has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this presentation level.</td>
</tr>
<tr>
<td>containerRef</td>
<td>References the presentation hierarchy to which this presentation level belongs.</td>
</tr>
<tr>
<td>logicalLevelRef</td>
<td>References the logical level upon which this presentation level is based.</td>
</tr>
</tbody>
</table>

The presentation level tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Lists any aliases that have been defined for this presentation level. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias has an attribute called name.</td>
</tr>
<tr>
<td>RefDrillTo</td>
<td>References the levels for cross-dimension drill-down operations, as in the case of a Region dimension with a City level that can have a &quot;drill to&quot; level Year in the Time dimension.</td>
</tr>
<tr>
<td>RefDisplayColumns</td>
<td>References the columns that have been designated to be used for display for this presentation level (on drill-down). Each column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>This child element is not used.</td>
</tr>
</tbody>
</table>

Syntax

<pre><PresentationLevel_</pre>
Example

```xml
<PresentationLevel mdsid="m80de558f-0fbe-0000-714b-e31d00000000"
name="Position Type"
hasDispName="false"
hasDispDescription="false"
containerRef="/oracle/bi/server/base/PresentationHierarchy/HR Positions_80de558d-0fbc-0000-714b-e31d00000000.xml#m80de558d-0fbc-0000-714b-e31d00000000"
logicalLevelRef="/oracle/bi/server/base/LogicalLevel/Position Type_002e58f1-07e9-0000-714b-e31d00000000.xml#m002e58f1-07e9-0000-714b-e31d00000000">
<RefDisplayColumns>
  <RefPresentationColumn refId="m80de558f-0fbe-0000-714b-e31d00000000-m80de5584-0faa-0000-714b-e31d00000000"
presentationColumnRef="/oracle/bi/server/base/PresentationColumn/HR Position_80de558c-0fa8-0000-714b-e31d00000000.xml#m80de5584-0faa-0000-714b-e31d00000000" />
  <RefPresentationColumn refId="m80de558f-0fbe-0000-714b-e31d00000000-m80de5585-0faa-0000-714b-e31d00000000"
presentationColumnRef="/oracle/bi/server/base/PresentationColumn/HR Position_80de558c-0fa8-0000-714b-e31d00000000.xml#m80de5585-0faa-0000-714b-e31d00000000" />
</RefDisplayColumns>
</PresentationLevel>
```
This chapter provides MDS XML reference information for Oracle Marketing Segmentation elements, security elements, and miscellaneous Oracle BI repository elements that do not fall in other categories.

This chapter contains the following topics:

- Section 5.1, "MDS XML Marketing Elements"
- Section 5.2, "MDS XML Security Elements"
- Section 5.3, "Other MDS XML Elements"

### 5.1 MDS XML Marketing Elements

Marketing elements are typically those elements that are related to features for the Oracle Marketing Segmentation product. For information about features and options related to Oracle Marketing Segmentation, see the administrator documentation for the Oracle Marketing Segmentation application.

Marketing elements include:

- Section 5.1.1, "MDS XML Elements: Catalog Link"
- Section 5.1.2, "MDS XML Elements: Target Level"
- Section 5.1.3, "MDS XML Elements: List Catalog"
- Section 5.1.4, "MDS XML Elements: Qualified List Item"
- Section 5.1.5, "MDS XML Elements: Qualifying Key"
- Section 5.1.6, "MDS XML Elements: Sampling Table"
- Section 5.1.7, "MDS XML Elements: Seg Catalog"

#### 5.1.1 MDS XML Elements: Catalog Link

The catalog link element corresponds to the conforming dimension link object in the Marketing metadata. A conforming dimension links a fact that contains target-level IDs to a fact that does not contain target-level IDs by navigating along a dimension that is shared by both fact tables.

The catalog link tag includes the following attributes:
Table 5–1  Catalog Link Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inColumnRef</td>
<td>References the presentation column that represents the primary key of the subject area from which the dimension link originates.</td>
</tr>
<tr>
<td>outColumnRef</td>
<td>References the presentation column that represents the primary key of the subject area to which the dimension link points.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<CatalogLink ...
    inColumnRef=""
    outColumnRef=""/>
<Description />
</CatalogLink>
```

Example

```xml
<CatalogLink mdsid="m8033142f-0fac-0000-714b-e31d00000000"
    name="Customer Profile->Opportunity Contact"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    inColumnRef="/oracle/bi/server/base/PresentationTable/_u0150 Contact Profile_802b468d-0fa8-0000-714b-e31d00000000.xml#m80312fa3-0faa-0000-714b-e31d00000000"
    outColumnRef="/oracle/bi/server/base/PresentationTable/_u0150 Contact Profile_80330ba0-0fa8-0000-714b-e31d00000000.xml#m80330b9e-0faa-0000-714b-e31d00000000"
>
    <Description>
        <![CDATA[ Used by Account Target Level to navigate to Opportunity Contact ]]>  
    </Description>
</CatalogLink>
```

5.1.2 MDS XML Elements: Target Level

The target level element corresponds to the target level object in the Marketing metadata. A target level is the entity that a marketer wants to count, typically customer types like individuals, businesses, or households.

The target level tag includes the following attributes:

Table 5–2  Target Level Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this target level has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this target level.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this target level has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this target level.</td>
</tr>
<tr>
<td>primaryQLIRef</td>
<td>References the primary qualified list item for this target level.</td>
</tr>
<tr>
<td>primaryCatalogRef</td>
<td>References the primary segmentation catalog for this target level.</td>
</tr>
<tr>
<td>staticSegmentCatalogRef</td>
<td>References the set of segmentation catalogs for this target level.</td>
</tr>
<tr>
<td>staticSegmentTargetColumnRef</td>
<td>References the Target Id presentation column.</td>
</tr>
</tbody>
</table>
The target level tag includes the following child elements:

**Table 5–3 Target Level Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeaderSQL</td>
<td>Contains the physical SQL to insert the Saved Result Header. Used when users save a result set.</td>
</tr>
<tr>
<td>DelHeaderSQL</td>
<td>Contains the physical SQL to delete the saved result set header. Used when users purge saved result sets.</td>
</tr>
<tr>
<td>DelDataSQL</td>
<td>Contains the physical SQL to delete the saved result data set. Used when users purge saved result sets.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<TargetLevel ...>
  <Description />  <!-- Description element -->
  <HeaderSQL ... /> <!-- HeaderSQL element -->
  <DelHeaderSQL ... /> <!-- DelHeaderSQL element -->
  <DelDataSQL ... />  <!-- DelDataSQL element -->
</TargetLevel>
```

**Example**

```xml
<TargetLevel mdsid="m802ba17b-0fae-0000-714b-e31d00000000"
  name="Contacts"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  hasDispName="false"
  primaryQLIRef="/oracle/bi/server/base/QualifiedListItem/Contacts_802ba17c-0fb2-0000-714b-e31d00000000.xml#m802ba17c-0fb2-0000-714b-e31d00000000"
  primaryCatalogRef="/oracle/bi/server/base/PresentationCatalog/Customer Profile_segmentation_802b44df-0fa4-0000-714b-e31d00000000.xml#m802b44df-0fa4-0000-714b-e31d00000000"
  staticSegmentTargetColumnRef="/oracle/bi/server/base/PresentationTable/_u0150 Contact Results Data_802ba2dd1-0fa8-0000-714b-e31d00000000.xml#m802ba2cd-0fa8-0000-714b-e31d00000000"
  staticSegmentNodeColumnRef="/oracle/bi/server/base/PresentationTable/_u0150"
  staticSegmentPhysicalTableRef="/oracle/bi/server/base/PresentationTable/_u0150"
  staticSegmentConnectionPoolRef="/oracle/bi/server/base/PresentationTable/_u0150"
</TargetLevel>
```
Contact Results Data_802ba2d1-0fa8-0000-714b-e31d00000000.xml#m802ba2cf-0faa-0000-714b-e31d00000000
staticSegmentPhysicalTableRef="/oracle/bi/server/base/PhysicalTable/M_SR_CONTACT_802ba200-0bb9-0000-714b-e31d00000000.xml#m802ba200-0bb9-0000-714b-e31d00000000"

Contact Results Data_802ba2d1-0fa8-0000-714b-e31d00000000.xml#m802ba2cf-0faa-0000-714b-e31d00000000
staticSegmentPhysicalTableRef="/oracle/bi/server/base/PhysicalTable/M_SR_CONTACT_802ba200-0bb9-0000-714b-e31d00000000.xml#m802ba200-0bb9-0000-714b-e31d00000000"

5.1.3 MDS XML Elements: List Catalog

The list catalog element corresponds to the list catalog object in the Marketing metadata. A list catalog is an object that is used to generate a list of data at the target level.

The list catalog key tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>presentationCatalogRef</td>
<td>References the subject area for this list catalog. The name of the subject area is used as the name of the list catalog.</td>
</tr>
</tbody>
</table>

The list catalog tag does not include any child elements.

Syntax

```xml
<ListCatalog ...
    presentationCatalogRef=""/>
</Description />
</ListCatalog>
```

Example

```xml
<ListCatalog mdsid="m8032fe28-0fb0-0000-714b-e31d00000000"
            name="Marketing Contact List"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns="http://www.oracle.com/obis/repository"
            presentationCatalogRef="/oracle/bi/server/base/PresentationCatalog/MktgContact List_8031263b-0fa4-0000-714b-e31d00000000.xml#m8031263b-0fa4-0000-714b-e31d00000000"/>
</ListCatalog>
```

5.1.4 MDS XML Elements: Qualified List Item

The qualified list item element corresponds to the qualified list item object in the Marketing metadata. A qualified list item is an entity that is evaluated against segment criteria so that the information related to that entity can be exported in a list file.

Table 5–4 List Catalog Tag Attributes
The qualified list item tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacheTable</td>
<td>Indicates whether this qualified list item is cached.</td>
</tr>
<tr>
<td>cacheCatalogRef</td>
<td>References the subject area for the cache for this qualified list item.</td>
</tr>
<tr>
<td>cacheConnPoolRef</td>
<td>References the connection pool for the cached table.</td>
</tr>
<tr>
<td>cacheTargetColumnRef</td>
<td>References the Target Id presentation column.</td>
</tr>
<tr>
<td>cacheNodeColumnRef</td>
<td>References the GUID presentation column.</td>
</tr>
<tr>
<td>cachePhysicalTableRef</td>
<td>References the physical table that stores the cached qualified list item.</td>
</tr>
</tbody>
</table>

The qualified list item tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteSQL</td>
<td>Contains the physical SQL used to delete the cached results.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<QualifiedListItem _
cacheTable=""
cacheCatalogRef=""
cacheConnPoolRef=""
cacheTargetColumnRef=""
cacheNodeColumnRef=""
cachePhysicalTableRef="">
<DeleteSQL>
<![CDATA[ [...] ]]>  
</DeleteSQL>
</QualifiedListItem>
```

Example

```xml
<QualifiedListItem mdsid="m80331498-0fb2-0000-714b-e31d00000000" name="Ordered Products"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.oracle.com/obis/repository"
cacheTable="M_C_GENERIC"
cacheCatalogRef="/oracle/bi/server/base/PresentationCatalog/Marketing Segmentation Cache and Saved Results_802ba2ad-0fa4-0000-714b-e31d00000000.xml#m802ba2ad-0fa4-0000-714b-e31d00000000"
cacheConnPoolRef="/oracle/bi/server/base/ConnectionPool/Oracle Data Warehouse Connection Pool_00238434-0bd5-0000-714b-e31d00000000.xml#m00238434-0bd5-0000-714b-e31d00000000"
cacheTargetColumnRef="/oracle/bi/server/base/PresentationTable/_u0150 Generic Item Cache_80331490-0fa8-0000-714b-e31d00000000.xml#m80331490-0fa8-0000-714b-e31d00000000"
cacheNodeColumnRef="/oracle/bi/server/base/PresentationTable/_u0150 Generic Item Cache_80331490-0fa8-0000-714b-e31d00000000.xml#m80331490-0fa8-0000-714b-e31d00000000"
cachePhysicalTableRef="/oracle/bi/server/base/PhysicalTable/M_C_GENERIC_80331468-0bb9-0000-714b-e31d00000000.xml#m80331468-0bb9-0000-714b-e31d00000000">
<DeleteSQL>
<![CDATA[ [...] ]]>  
</DeleteSQL>
</QualifiedListItem>
```
5.1.5 MDS XML Elements: Qualifying Key

The qualifying key element corresponds to the qualifying keys that are used in list catalog or segmentation catalog objects in the Marketing metadata. Qualifying keys map qualified list items to presentation columns.

The qualifying key tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listCatalogRef</td>
<td>References the list catalog to which this qualifying key belongs.</td>
</tr>
<tr>
<td>segmentationCatalogRef</td>
<td>References the segmentation catalog to which this qualifying key belongs.</td>
</tr>
<tr>
<td>keyColumnRef</td>
<td>References the presentation column for this qualifying key.</td>
</tr>
<tr>
<td>qualifiedListItemRef</td>
<td>References the qualified list item for this qualifying key.</td>
</tr>
</tbody>
</table>

**Syntax**

```
<QualifyingKey ...
    listCatalogRef=""
    segmentationCatalogRef=""
    keyColumnRef=""
    qualifiedListItemRef=""
    </Description />
</QualifyingKey>
```

**Example**

```
<QualifyingKey mdsid="m80c10f04-0fb4-0000-714b-e31d00000000"
    name="MKTG_QK_4020:4722171188242192"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    segmentationCatalogRef="/oracle/bi/server/base/SegCatalog/Accounts_Pharma Sales Effectiveness_Segmentation_80c10f0a-0fba-0000-714b-e31d00000000.xml#m80c10f0a-0fba-0000-714b-e31d00000000"
    keyColumnRef="/oracle/bi/server/base/PresentationTable/Accounts_80c0ef0d-0fa8-0000-714b-e31d00000000.xml#m80c0ef09-0faa-0000-714b-e31d00000000"
    qualifiedListItemRef="/oracle/bi/server/base/QualifiedListItem/Accounts_802bab1a6-0fb2-0000-714b-e31d00000000.xml#m802bab1a6-0fb2-0000-714b-e31d00000000">
</QualifyingKey>
```

5.1.6 MDS XML Elements: Sampling Table

The sampling table element corresponds to the sampling table object in the Marketing metadata. Sampling tables are created and managed as part of target level objects.

The sampling table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>factor</td>
<td>The sampling factor (for example, 10%).</td>
</tr>
<tr>
<td>targetLevelRef</td>
<td>References the target level for the sampling table.</td>
</tr>
</tbody>
</table>
Table 5–8 (Cont.) Sampling Table Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>physicalTableRef</td>
<td>References the physical table being used as the sampling table.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<SamplingTable
  factor=""
  targetLevelRef=""
  physicalTableRef="">
  <Description />
</SamplingTable>
```

Example

```xml
<SamplingTable mdsid="m80342e77-0fb8-0000-714b-e31d00000000"
  name="M_10A_ASSET_D"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  factor="0.1"
  targetLevelRef="/oracle/bi/server/base/TargetLevel/Assets_80341f24-0fae-0000-714b-e31d00000000.xml#m80341f24-0fae-0000-714b-e31d00000000"
  physicalTableRef="/oracle/bi/server/base/PhysicalTable/W_ASSET_D_00226101-0bb9-0000-714b-e31d00000000.xml#m00226101-0bb9-0000-714b-e31d00000000">
  <Description />
</SamplingTable>
```

5.1.7 MDS XML Elements: Seg Catalog

The seg catalog element corresponds to the segmentation catalog object in the Marketing metadata. Segmentation catalogs are Oracle BI subject areas that are enabled for segmentation. Segmentation catalogs provides a set of dimensions and fact measures that can be used to create segment criteria.

The seg catalog tag includes the following attributes:

Table 5–9 Seg Catalog Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>targetLevelRef</td>
<td>References the target level for the segmentation catalog.</td>
</tr>
<tr>
<td>presentationCatalogRef</td>
<td>References the subject area for this segmentation catalog.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<SegCatalog
  targetLevelRef=""
  presentationCatalogRef="">
  <Description />
</SegCatalog>
```

Example

```xml
<SegCatalog mdsid="m802ba19b-0fba-0000-714b-e31d00000000"
  name="Contacts_Quotes_segmentation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  targetLevelRef="/oracle/bi/server/base/TargetLevel/Contacts_802ba17b-0fae-0000-714b-e31d00000000.xml#m802ba17b-0fae-0000-714b-e31d00000000"
  presentationCatalogRef="/oracle/bi/server/base/PresentationCatalog/Quotes_segmentation_802b5904-0fa4-0000-714b-e31d00000000.xml#m802b5904-0fa4-0000-714b-e31d00000000">
  <Description />
</SegCatalog>
```
5.2 MDS XML Security Elements

Security elements are typically those elements that are related to Oracle Business Intelligence security features, such as those related to variables, authentication, privileges, users, and application roles. Security elements include:

- Section 5.2.1, "MDS XML Elements: Variable"
- Section 5.2.2, "MDS XML Elements: LDAP Server"
- Section 5.2.3, "MDS XML Elements: Init Block"
- Section 5.2.4, "MDS XML Elements: Custom Authenticator"
- Section 5.2.5, "MDS XML Elements: Privilege Package"
- Section 5.2.6, "MDS XML Elements: Object Privilege"
- Section 5.2.7, "MDS XML Elements: Query Privilege"
- Section 5.2.8, "MDS XML Elements: User"
- Section 5.2.9, "MDS XML Elements: Group"
- Section 5.2.10, "MDS XML Elements: DB Sign-On"

5.2.1 MDS XML Elements: Variable

The variable element corresponds to the variable object in the Physical layer of the repository.

The variable tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>securityMask</td>
<td>For session variables only. This attribute stores the value of the following options:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Enable any user to set the value.</strong> When selected, lets session variables be set after the initialization block has populated the value (at user login) by calling the ODBC store procedure <code>NQSSetSessionValue()</code>. For example, this option lets non-administrators set this variable for sampling.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Security Sensitive.</strong> When selected, identifies the variable as sensitive to security for virtual private databases (VPDs). When filtering cache table matches, the Oracle BI Server looks at the parent database of each column or table that is referenced in the logical request projection list. If the physical database source is a VPD, the Oracle BI Server matches a list of security-sensitive variables to each prospective cache hit. Cache hits would only occur on cache entries that included and matched all security-sensitive variables.</td>
</tr>
<tr>
<td>isSessionVar</td>
<td>If set to true, indicates that this variable is a session variable. Session variables are created and assigned a value when each user logs on, unlike repository variables, which have static values.</td>
</tr>
<tr>
<td>initBlockRef</td>
<td>References the init block where this variable is initialized.</td>
</tr>
</tbody>
</table>
The variable tag includes the following child elements:

### Table 5–11 Variable Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expr</td>
<td>Contains the expression that forms the definition of this variable.</td>
</tr>
</tbody>
</table>

### Syntax

```xml
<Variable name="" securityMask="" isSessionVar="" initBlockRef="">
  <Description />
  <Expr />
    <ExprText>...</ExprText>
    <ExprTextDesc>...</ExprTextDesc>
    <ObjectRefList>
      <RefObject objectRef="" .../>
    </ObjectRefList>
  </Expr>
</Variable>
```

### Example

```xml
<Variable mdsid="m804054aa-0bd7-0000-714b-e31d00000000"
  name="CURRENT_DAY"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository">
  <Description>
    <![CDATA[ Returns the value of Current Date in the MM/DD/YYYY format. ]]]>
  </Description>
  <Expr>
    <ExprText>
      <![CDATA[ VALUEOF("%1") ]]]>
    </ExprText>
    <ExprTextDesc>
      <![CDATA[ VALUEOF("Warehouse Refresh Date"."LAST_REFRESH_DT") ]]]>
    </ExprTextDesc>
    <ObjectRefList>
      <RefObject refId="m804054aa-0bd7-0000-714b-e31d00000000-m0023848c-0bd7-0000-714b-e31d00000000" objectRef="/oracle/bi/server/base/Variable/LAST_REFRESH_DT_0023848c-0bd7-0000-714b-e31d00000000.xml#m0023848c-0bd7-0000-714b-e31d00000000"/>
    </ObjectRefList>
  </Expr>
</Variable>
```

### 5.2.2 MDS XML Elements: LDAP Server

The LDAP server element corresponds to the LDAP server object in the Identity Manager dialog in the Administration Tool.

The LDAP server tag includes the following attributes:
The LDAP server tag does not include any child elements.

### Syntax
```
<LDAPServer ...
    hostName=""  
    port=""    
    baseDN=""    
    bindDN=""    
    userNameAttrType=""    
    domainID=""    
    password=""    
    timeout=""    
    cacheExpiry=""    
    isADSI=""    
    useSSL=""    
    keyFile=""    
    keyPassword=""    
    version=""    
    cacheEntries="" />
```

### Table 5–12 LDAP Server Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostName</td>
<td>The host name of the LDAP server.</td>
</tr>
<tr>
<td>port</td>
<td>The port number of the LDAP server (for example, 389).</td>
</tr>
<tr>
<td>baseDN</td>
<td>The base distinguished name (DN) that identifies the starting point of the authentication search.</td>
</tr>
<tr>
<td>bindDN</td>
<td>The optional DN that is required to bind to the LDAP server. This option is required if <code>isADSI</code> is set to true.</td>
</tr>
<tr>
<td>userNameAttrType</td>
<td>A value that uniquely identifies a user, such as the user ID, the RDN (relative distinguished name), or (for ADSI) sAMAccountName.</td>
</tr>
<tr>
<td>domainID</td>
<td>A domain identifier that uniquely identifies the domain for which the LDAP object is responsible. This is especially useful when there are multiple LDAP objects.</td>
</tr>
<tr>
<td>password</td>
<td>The user password associated with the Bind DN. This option is required if <code>isADSI</code> is set to true. Note that passwords are encrypted.</td>
</tr>
<tr>
<td>timeout</td>
<td>The time period after which Administration Tool import attempts or Oracle BI Server user authentication against this LDAP server will time out.</td>
</tr>
<tr>
<td>cacheExpiry</td>
<td>The interval at which the authentication cache entry in the LDAP cache for a logged on user is refreshed. This option corresponds to <strong>Cache refresh interval</strong> in the Repository tab of the Options dialog in the Administration Tool.</td>
</tr>
<tr>
<td>isADSI</td>
<td>If set to true, indicates that this is an Active Directory LDAP server.</td>
</tr>
<tr>
<td>useSSL</td>
<td>If set to true, indicates that the connection to the LDAP server uses Secure Sockets Layer (SSL).</td>
</tr>
<tr>
<td>keyFile</td>
<td>The name of the key file that holds the client certificate and Certificate Authority (CA) certificate. Required if <code>useSSL</code> is set to true. This option corresponds to <strong>Key file name</strong> in the Repository tab of the Options dialog in the Administration Tool.</td>
</tr>
<tr>
<td>keyPassword</td>
<td>The password that is required to connect to the external LDAP server.</td>
</tr>
<tr>
<td>version</td>
<td>The LDAP version number of the LDAP server, either 2 or 3. The default is 3.</td>
</tr>
<tr>
<td>cacheEntries</td>
<td>The maximum number of entries in the authentication cache, preallocated when the Oracle BI Server starts. If the number of users exceeds this limit, cache entries are replaced using the LRU algorithm. If this value is 0, then the authentication cache is disabled. This option corresponds to <strong>Number of Cache Entries</strong> in the Repository tab of the Options dialog in the Administration Tool.</td>
</tr>
</tbody>
</table>
Example

```xml
<LDAPServer mdsid="mbd837790-0bf6-1000-806e-0a97078d0000"
    name="LDAP01"
    isDefaultReadable="false"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    hostName="sdacns05"
    port="389"
    version="3"
    baseDN="name1"
    bindDN="name2"
    userNameAttrType="sAMAccountName"
    password="6C6F0BE01FC4111F3AC2236A364D912EBBFE551E0A2AC2F5352A18F9228901AA7D896E3DA56EBC22P248B15B1759E26A638B148E7051643B2E72CAAC984A64515B58DEA74F85662056F7952065929ADEF"%
    timeout="60"
    cacheExpiry="86400"
    cacheEntries="1000"
    isADSI="true"
    useSSL="false"
    keyPassword="E3130008E1C4CAD47041E4AE6B04867C2E35213306F12832914CBE7A9DD95561D771DEDD64A84112B1FC627B6D0D58">%>
</LDAPServer>
```

5.2.3 MDS XML Elements: Init Block

The init block element corresponds to the initialization block object in the Physical layer. Initialization blocks are used to initialize dynamic repository variables, system session variables, and nonsystem session variables.

The init block tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isSessionVar</td>
<td>If set to true, indicates that this initialization block is a session initialization block. Session initialization blocks are used with session variables. The default value is false.</td>
</tr>
</tbody>
</table>
The init block tag includes the following child elements:

Table 5–14 Init Block Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMapItem</td>
<td>Stores the value of database-specific strings. The <code>&lt;DBMapItem&gt;</code> element contains a <code>&lt;name&gt;</code> attribute and a <code>&lt;Value&gt;</code> sub-element.</td>
</tr>
</tbody>
</table>
**Table 5-14 (Cont.) Init Block Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefPredecessors</td>
<td>References the initialization blocks, if any, that are initialized before this initialization block. Each initialization block is contained in a sub-element called RefInitBlock.</td>
</tr>
<tr>
<td>RefVariables</td>
<td>References the variables that are associated with this initialization block. Each variable is contained in a sub-element called RefVariable.</td>
</tr>
<tr>
<td>InitString</td>
<td>The SQL initialization string needed to populate the variables. This element is only used if the data source type for this initialization block is Database or XML.</td>
</tr>
<tr>
<td>RefLDAPServers</td>
<td>References the LDAP servers that are associated with this initialization block. Each LDAP server is contained in a sub-element called RefLDAPServer. This element is only used if the data source type for this initialization block is LDAP.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<InitBlock ... isSessionVar="" refreshPeriod="" baseTime="" isRowWiseInit="" isRowWiseCacheEnabled="" isSuccRequired="" isEnabled="" isDeferredExecution="" isDBIndependentSQL="" connectionPoolRef="">  
  <Description />  
  <DBMapItem name="">  
    <Value .../>  
  </DBMapItem>  
  <RefPredecessors>  
    <RefInitBlock predecessorRef="" ...""/>  
    <RefInitBlock predecessorRef="" ...""/>  
  </RefPredecessors>  
  <RefVariables>  
    <RefVariable variableRef="" ...""/>  
    <RefVariable variableRef="" ...""/>  
  </RefVariables>  
  <InitString .../>  
  <RefLDAPServers>  
    <RefLDAPServer ldapServerRef="" ...""/>  
    <RefLDAPServer ldapServerRef="" ...""/>  
  </RefLDAPServers>  
</InitBlock>
```

**Example**

```xml
<InitBlock mdsid="m002384be-0bd9-0000-714b-e31d00000000" name="Primary Owner ID" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.oracle.com/obis/repository" ...
```
isSessionVar="true"
isRowWiseCacheEnabled="false"
isEnabled="false"
connectionPoolRef="/oracle/bi/server/base/ConnectionPool/Siebel OLTP Connection Pool_00238431-0bd5-0000-714b-e31d00000000.xml#m00238431-0bd5-0000-714b-e31d00000000">
<DBMapItem name="DefaultMulDB">
<Value>
<![CDATA[
    select PAR_ROW_ID
    from VALUEOF(TBO).S_USER
    where LOGIN = ':USER' ]]>}
</Value>
</DBMapItem>
<RefPredecessors>
<RefInitBlock refId="m002384be-0bd9-0000-714b-e31d00000000-m002384ac-0bd9-0000-714b-e31d00000000"
  predecessorRef="/oracle/bi/server/base/InitBlock/Authentication_002384ac-0bd9-0000-714b-e31d00000000.xml#m002384ac-0bd9-0000-714b-e31d00000000" />
</RefPredecessors>
<RefVariables>
<RefVariable refId="m002384be-0bd9-0000-714b-e31d00000000-m0023845e-0bd7-0000-714b-e31d00000000"
  variableRef="/oracle/bi/server/base/Variable/PR_OWNER_ID_0023845e-0bd7-0000-714b-e31d00000000.xml#m0023845e-0bd7-0000-714b-e31d00000000" />
</RefVariables>
<InitString>
<![CDATA[
    select PAR_ROW_ID
    from VALUEOF(TBO).S_USER
    where LOGIN = ':USER' ]]>}
</InitString>
</InitBlock>

5.2.4 MDS XML Elements: Custom Authenticator

The Custom Authenticator element corresponds to the Custom Authenticator object in the Identity Manager dialog in the Administration Tool.

The Custom Authenticator tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacheExpire</td>
<td>The interval at which the authentication cache entry for a logged on user is refreshed for this custom authenticator.</td>
</tr>
<tr>
<td>path</td>
<td>The path and name of the authenticator plug-in dll.</td>
</tr>
<tr>
<td>configParam</td>
<td>Lists any parameters for this custom authenticator that have been explicitly exposed for configuration.</td>
</tr>
<tr>
<td>hiddenParam</td>
<td>Lists any parameters for this custom authenticator that have been encrypted, such as passwords.</td>
</tr>
<tr>
<td>cacheEntries</td>
<td>The maximum number of entries in the authentication cache for this custom authenticator, preallocated when the Oracle BI Server starts. If the number of users exceeds this limit, cache entries are replaced using the LRU algorithm. If this value is 0, then the authentication cache is disabled.</td>
</tr>
<tr>
<td>csfKey</td>
<td>This attribute is not used.</td>
</tr>
</tbody>
</table>
The Custom Authenticator element does not include any child elements.

Syntax

```xml
<CustomAuthenticator ...
  isDefaultReadable=""
  cacheExpiry=""
  path=""
  configParam=""
  hiddenParam=""
  cacheEntries=""
  initBlockRef="">

  <Description />

</CustomAuthenticator>
```

Example

```xml
<CustomAuthenticator mdsid="mbdf488e1-0bf6-1000-806e-0a97078d0000"
  name="CA01"
  isDefaultReadable="false"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository"
  cacheExpiry='86400'
  cacheEntries='1000'
  path="authenticate.dll"
  configParam='ABC'
  hiddenParam="D7EDED84BC624A917F5B462A4DCA05CDCE256EEEEEDC97D5D885270569A16AB6">

  <Description />

</CustomAuthenticator>
```

5.2.5 MDS XML Elements: Privilege Package

The privilege package element is a container for other privileges. It corresponds to a set of permissions that applies to a particular user or application role object in the repository metadata. The privilege package element contains the information available on the Object Permissions, Data Filters, and Query Limits tabs of the User/Application Role Permissions dialog in the Administration Tool.

Because privilege packages can be associated with more than one user or application role, the total number of privilege packages may not equal the total number of users and application roles.

The privilege package tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userRef</td>
<td>References any users that have the privilege package.</td>
</tr>
<tr>
<td>roleRef</td>
<td>References any application roles that have the privilege package.</td>
</tr>
<tr>
<td>isFineGrainedMode</td>
<td>Reserved for Oracle internal use only.</td>
</tr>
</tbody>
</table>

The privilege package tag includes the following child elements:
Table 5–17 Privilege Package Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryPrivilegeMapping</td>
<td>References any query privileges that belong to this privilege package, along with the corresponding database. Contains two attributes: databaseRef, which references the database name, and queryPrivilegeRef, which references the query privilege.</td>
</tr>
<tr>
<td>FilterMapping</td>
<td>References any data filters that belong to this privilege package by mapping an object name with the filter expression. Contains the following attributes and sub-elements: Status. Valid values are Enabled, Disabled, and Ignored, objectRef, which references the metadata object name, Expr, which contains the filter expression</td>
</tr>
</tbody>
</table>

Syntax

```
<PrivilegePackage ... userRef="" roleRef="" isFineGrainedMode="">
  <Description />
  <QueryPrivilegeMapping databaseRef="" queryPrivilegeRef="" />
  <FilterMapping status="" objectTypeID="" objectRef="">
    <Expr>
      <ExprText>...</ExprText>
    </ExprText>
    <ExprTextDesc>...</ExprTextDesc>
    <ObjectRefList>
      <RefObject objectRef="" />
    </ObjectRefList>
  </Expr>
</FilterMapping>
</PrivilegePackage>
```

Example

```
<PrivilegePackage mdsid="m808626e7-1071-0000-714b-e31d00000000" name="PrivPack_4209:3417381144981469" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.oracle.com/obis/repository" roleRef="/oracle/bi/server/base/Group/Purchasing_Buyer_0041592c-106e-0000-714b-e31d00000000.xml#m0041592c-106e-0000-714b-e31d00000000">
  <QueryPrivilegeMapping databaseRef="/oracle/bi/server/base/Databse/Oracle Data Warehouse_0023840a-0bcf-0000-714b-e31d00000000.xml#m0023840a-0bcf-0000-714b-e31d00000000" queryPrivilegeRef="/oracle/bi/server/base/QueryPrivilege/QueryPriv_4204_u0053_3417391144981469_808626e8-106c-0000-714b-e31d00000000.xml#m808626e8-106c-0000-714b-e31d00000000" />
  <FilterMapping status="disable" objectTypeID='2035' objectRef="/oracle/bi/server/base/LogicalTable/Fact_80d105ed-07f3-0000-714b-e31d00000000.xml#m80d105ed-07f3-0000-714b-e31d00000000" />
```

Table 5–17 Privilege Package Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryPrivilegeMapping</td>
<td>References any query privileges that belong to this privilege package, along with the corresponding database. Contains two attributes: databaseRef, which references the database name, and queryPrivilegeRef, which references the query privilege.</td>
</tr>
<tr>
<td>FilterMapping</td>
<td>References any data filters that belong to this privilege package by mapping an object name with the filter expression. Contains the following attributes and sub-elements: Status. Valid values are Enabled, Disabled, and Ignored, objectRef, which references the metadata object name, Expr, which contains the filter expression</td>
</tr>
</tbody>
</table>

Syntax

```
<PrivilegePackage ... userRef="" roleRef="" isFineGrainedMode="">
  <Description />
  <QueryPrivilegeMapping databaseRef="" queryPrivilegeRef="" />
  <FilterMapping status="" objectTypeID="" objectRef="">
    <Expr>
      <ExprText>...</ExprText>
    </ExprText>
    <ExprTextDesc>...</ExprTextDesc>
    <ObjectRefList>
      <RefObject objectRef="" />
    </ObjectRefList>
  </Expr>
</FilterMapping>
</PrivilegePackage>
```

Example

```
<PrivilegePackage mdsid="m808626e7-1071-0000-714b-e31d00000000" name="PrivPack_4209:3417381144981469" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.oracle.com/obis/repository" roleRef="/oracle/bi/server/base/Group/Purchasing_Buyer_0041592c-106e-0000-714b-e31d00000000.xml#m0041592c-106e-0000-714b-e31d00000000">
  <QueryPrivilegeMapping databaseRef="/oracle/bi/server/base/Databse/Oracle Data Warehouse_0023840a-0bcf-0000-714b-e31d00000000.xml#m0023840a-0bcf-0000-714b-e31d00000000" queryPrivilegeRef="/oracle/bi/server/base/QueryPrivilege/QueryPriv_4204_u0053_3417391144981469_808626e8-106c-0000-714b-e31d00000000.xml#m808626e8-106c-0000-714b-e31d00000000" />
  <FilterMapping status="disable" objectTypeID='2035' objectRef="/oracle/bi/server/base/LogicalTable/Fact_80d105ed-07f3-0000-714b-e31d00000000.xml#m80d105ed-07f3-0000-714b-e31d00000000" />
```
<Expr>
  <ExprText>
    <![CDATA[ cast("%1" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG) AND ("%2" = valueof(NQ_SESSION."%3") OR cast("%1" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG_ViewOthers)) ]]></ExprText>
  <ExprTextDesc>
    <![CDATA[ cast("Core":"Dim - Procurement Business Unit","Key Id" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG) AND ("Core":"Dim - Purchase Buyer","Purchase Buyer Login" = valueof(NQ_SESSION."USER") OR cast("Core":"Dim - Procurement Business Unit","Key Id" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG_ViewOthers)) ]]></ExprTextDesc>
</Expr>
</FilterMapping>
<FilterMapping status="disable" objectTypeId="2035"
  objectRef="/oracle/bi/server/base/LogicalTable/Fact _u0150 Sourcing _u0150 Negotiation Award_80d105fe-07f3-0000-714b-e31d00000000.xml#m80d105fe-07f3-0000-714b-e31d00000000">
  <Expr>
    <ExprText>
      <![CDATA[ cast("%1" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG) AND ("%2" = valueof(NQ_SESSION."%3") OR cast("%1" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG_ViewOthers)) ]]></ExprText>
    <ExprTextDesc>
      <![CDATA[ cast("Core":"Dim - Procurement Business Unit","Key Id" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG) AND ("Core":"Dim - Purchase Buyer","Purchase Buyer Login" = valueof(NQ_SESSION."USER") OR cast("Core":"Dim - Procurement Business Unit","Key Id" AS DOUBLE) = valueof(NQ_SESSION.PRC_BU_MANAGE_NEG_ViewOthers)) ]]></ExprTextDesc>
  </Expr>
</FilterMapping>
</FilterMapping>
<RefObject refId="m808626e7-1071-0000-714b-e31d00000000-m000b17e8-07d6-0000-714b-e31d00000000"
  objectTypeId="2006"
  objectRef="/oracle/bi/server/base/LogicalTable/Dim _u0150 Procurement Business Unit_000b17df-07f3-0000-714b-e31d00000000.xml#m000b17e8-07d6-0000-714b-e31d00000000" />
<RefObject refId="m808626e7-1071-0000-714b-e31d00000000-m80ea6449-07d6-0000-714b-e31d00000000"
  objectTypeId="2006"
  objectRef="/oracle/bi/server/base/LogicalTable/Dim _u0150 Purchase Buyer_805fe01a-07f3-0000-714b-e31d00000000.xml#m80ea6449-07d6-0000-714b-e31d00000000" />
<RefObject refId="m808626e7-1071-0000-714b-e31d00000000-m00238445-0bd7-0000-714b-e31d00000000"
  objectTypeId="2006"
  objectRef="/oracle/bi/server/base/Variable/USER_00238445-0bd7-0000-714b-e31d00000000.xml#m00238445-0bd7-0000-714b-e31d00000000" />
</ObjectRefList>
5.2.6 MDS XML Elements: Object Privilege

The object privilege element corresponds to the set of object permissions you can apply to metadata objects such as logical or presentation columns. In the Administration Tool, you can set object permissions in the Presentation layer, or in the Object Permissions tab of the User/Application Role Permissions dialog.

The object privilege tag includes the following attributes:
The object privilege tag includes the following child elements:

### Table 5–18  Object Privilege Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Contains the type of privilege applied to the objects. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ read: corresponds to the <strong>Read</strong> privilege in the Administration Tool</td>
</tr>
<tr>
<td></td>
<td>■ readWrite: corresponds to the <strong>Read/Write</strong> privilege in the Administration Tool</td>
</tr>
<tr>
<td></td>
<td>■ none: corresponds to the <strong>No Access</strong> privilege in the Administration Tool</td>
</tr>
<tr>
<td></td>
<td>This attribute can contain the following other values: delete, write, grant, and all. These other values are for internal use.</td>
</tr>
<tr>
<td>privilegePackageRef</td>
<td>References the privilege package block for the object privilege.</td>
</tr>
</tbody>
</table>

### Syntax

```xml
<ObjectPrivilege ...
    type=""
    privilegePackageRef=""/>
<Description />
<RefObjects>
    <RefObject objectRef="" .../>
    <RefObject objectRef="" .../>
</RefObjects>
</ObjectPrivilege>
```

### Example

```xml
<ObjectPrivilege mdsid="m80343298-05dc-0000-714b-e31d00000000"
    name="ObjPriv_1500:1678341098389037"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository"
    type="none"
    privilegePackageRef="/oracle/bi/server/base/PrivilegePackage/PrivPack_4209_u0058_127804102157837_0023e21b-1071-0000-714b-e31d00000000.xml#m0023e21b-1071-0000-714b-e31d00000000"/>
<RefObjects>
    <RefObject refId="m80343298-05dc-0000-714b-e31d00000000-m80343156-0fa8-0000-714b-e31d00000000" objectTypeId="4008"
        objectRef="/oracle/bi/server/base/PresentationTable/Fact_u0150 Contact_u0150_80343156-0fa8-0000-714b-e31d00000000.xml#m80343156-0fa8-0000-714b-e31d00000000" />
    <RefObject refId="m80343298-05dc-0000-714b-e31d00000000-m8034314d-0fa8-0000-714b-e31d00000000" objectTypeId="4008"
        objectRef="/oracle/bi/server/base/PresentationTable/Fact_u0150 Contact_u0150_8034314d-0fa8-0000-714b-e31d00000000.xml#m8034314d-0fa8-0000-714b-e31d00000000" />
</RefObjects>
```
5.2.7 MDS XML Elements: Query Privilege

The query privilege element corresponds to the set of query privileges you can apply to users or application roles. In the Administration Tool, you can set query privileges in the Query Limits tab of the User/Application Role Permissions dialog.

The query privilege tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxExecTime</td>
<td>The maximum number of minutes queries can run on a database.</td>
</tr>
<tr>
<td>maxExecTimePrivilege</td>
<td>Specifies the status of the maximum query run time limit. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Enable: Limits the time to the value specified in maxExecTime.</td>
</tr>
<tr>
<td></td>
<td>■ Disable: Disables any limits set in maxExecTime.</td>
</tr>
<tr>
<td></td>
<td>■ Warn: Does not enforce limits, but logs queries that exceed the set time limit in the Query log.</td>
</tr>
<tr>
<td></td>
<td>■ Ignore: Inherits limits from the parent application role. If there is no time limit to inherit, no limit is enforced.</td>
</tr>
<tr>
<td>maxRows</td>
<td>The maximum number of rows for users to retrieve from a database.</td>
</tr>
<tr>
<td>maxRowsPrivilege</td>
<td>Specifies the status of the maximum number of rows limit. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Enable: Limits the number of rows to the value specified. If the number of rows exceeds the maxRows value, the query is terminated.</td>
</tr>
<tr>
<td></td>
<td>■ Disable: Disables any limits set in maxRows.</td>
</tr>
<tr>
<td></td>
<td>■ Warn: Does not enforce limits, but logs queries that exceed the set limit in the Query log.</td>
</tr>
<tr>
<td></td>
<td>■ Ignore: Inherits limits from the parent application role. If there is no row limit to inherit, no limit is enforced.</td>
</tr>
<tr>
<td>populatePrivilege</td>
<td>Specifies whether the Populate privilege is granted or denied for a database. The Populate stored procedure writes the Cache/Saved Result Set value to the database when a criteria block is cached. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Allow: Explicitly grants the Populate privilege.</td>
</tr>
<tr>
<td></td>
<td>■ Disallow: Explicitly denies the Populate privilege.</td>
</tr>
<tr>
<td></td>
<td>■ Ignore: Inherits limits from the parent application role. If there is no limit to inherit, then the Populate privilege is allowed or disallowed based on the property Allow populate queries by default for the database object.</td>
</tr>
</tbody>
</table>
The query privilege tag includes the following child elements:

**Table 5–21 Query Privilege Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecTimeTableItem</td>
<td>If database access has been restricted to particular time periods, lists the time periods and whether access is allowed or disallowed. Contains three attributes: allow (true or false), startTime, and endTime.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<QueryPrivilege _ 
  maxExecTime="" 
  maxExecTimePrivilege="" 
  maxRows="" 
  maxRowsPrivilege="" 
  populatePrivilege="" 
  execPhysicalPrivilege=""> 
  <Description /> 
  <ExecTimeTableItem allow="" startTime="" endTime="/>
</QueryPrivilege>
```

**Example**

```xml
<QueryPrivilege mdsid="m803ac151-106c-0000-714b-e31d00000000" 
  name='QueryPriv_4204:1926721120861104' 
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 
  xmlns='http://www.oracle.com/obis/repository' 
  maxExecTime='600' 
  maxRows='100000' 
  populatePrivilege='enable'> 
</QueryPrivilege>
```

### 5.2.8 MDS XML Elements: User

The user element corresponds to the user object in the repository. These objects represent the metadata repository copies or versions of externally managed user objects.

The user tag includes the following attributes:
The user tag includes the following child elements:

Table 5–22 User Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fullName</td>
<td>The full name of this user.</td>
</tr>
<tr>
<td>loggingLevel</td>
<td>The logging level for this user.</td>
</tr>
<tr>
<td>guid</td>
<td>Global unique identifier for a given user, which is independent of the user name.</td>
</tr>
</tbody>
</table>

Table 5–23 User Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefGroups</td>
<td>References the application roles to which this user belongs. Each application role is contained in a sub-element called RefGroup.</td>
</tr>
<tr>
<td>DBSignOnListMapping</td>
<td>References database login user names and passwords, along with the corresponding database, if this user has database-specific logins defined. Contains two attributes: dbSignOnRef, which references the database credentials, and databaseRef, which references the database name. Note that database passwords are encrypted.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<User ...
   fullName=""  
   loggingLevel=""  
   guid="">  
   <Description />  
   <RefGroups>  
     <RefGroup groupRef="" />  
     <RefGroup groupRef="" />  
   </RefGroups>  
   <DBSignOnListMapping dbSignOnRef="" databaseRef="" />  
</User>
```

Example

```xml
<User mdsid="m0021f847-05dd-0000-714b-e31d00000000"
     name="Administrator"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns="http://www.oracle.com/obis/repository"
     guid="B0E8200AB53A11DE8F4643DF3B7A9E29"
     loggingLevel="6">
   <RefGroups>
     <RefGroup refId="m0021f847-05dd-0000-714b-e31d00000000-m0023e1a2-106e-0000-714b-e31d00000000"
                      groupRef="/oracle/bi/server/base/Group/BIAdministrators_0023e1a2-106e-0000-714b-e31d00000000.xml#m0023e1a2-106e-0000-714b-e31d00000000"/>
     <RefGroup refId="m0021f847-05dd-0000-714b-e31d00000000-m0023e1a7-106e-0000-714b-e31d00000000"
                      groupRef="/oracle/bi/server/base/Group/Service Analytics Administrator_0023e1a7-106e-0000-714b-e31d00000000.xml#m0023e1a7-106e-0000-714b-e31d00000000"/>
   </RefGroups>
</User>
```
5.2.9 MDS XML Elements: Group

The group element corresponds to the application role object in the repository metadata.

The group tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayName</td>
<td>The custom Display Name for this application role.</td>
</tr>
<tr>
<td>logLevel</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>logStatisticsStatus</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>guid</td>
<td>Global unique identifier for a given application role, which is independent of the role name.</td>
</tr>
</tbody>
</table>

The group tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefChildGroups</td>
<td>References the set of application roles that belong to this application role. Each child application role is contained in a sub-element called RefGroup.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<Group ...
   displayName=""  
guid="" ... />
<Description ... />
<RefChildGroups>
   <RefGroup groupRef="" ... />
   <RefGroup groupRef="" ... />
</RefChildGroups>
</Group>
```

Example

```xml
<Group mdsid="m0023e1d2-106e-0000-714b-e31d00000000"
   name="Insurance"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.oracle.com/obis/repository">
   <RefChildGroups>
   ...<RefGroup refId="m0023e1d2-106e-0000-714b-e31d00000000-m0023e1dc-106e-0000-714b-e31d00000000" ... />
   </RefChildGroups>
</Group>
```
5.2.10 MDS XML Elements: DB Sign-On

The DB sign-on element corresponds to the information provided in the Logons tab of the user object. If the repository has been set up for database-specific login IDs and passwords, you can set up database-specific credentials to allow privileged users to connect to the underlying databases at the level of authority granted to those users in the databases.

The DB sign-on tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>The user name for the underlying database.</td>
</tr>
<tr>
<td>password</td>
<td>The corresponding password for the underlying database. Note that passwords are encrypted.</td>
</tr>
</tbody>
</table>

The DB sign-on tag does not include any child elements.

Syntax

```xml
<DBSignOn ...
    user="my_db_user"
    password="BA0EAAB163">
</DBSignOn>
```

Example

```xml
<DBSignOn mdsid="me406bac9-0bf1-1000-b290-0ae593200000" 
    name="LOGON_4211:255932074113" 
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 
    xmlns="http://www.oracle.com/obis/repository" 
    user="my_db_user"
    password="BA0EAAB163">
</DBSignOn>
```

5.3 Other MDS XML Elements

The elements described in this section are miscellaneous elements that do not fit into any other category. These elements include:

- Section 5.3.1, "MDS XML Elements: Project"
- Section 5.3.2, "MDS XML Elements: Query Object"

5.3.1 MDS XML Elements: Project

The project element corresponds to the project object in the repository metadata. Projects support multiuser development by enabling developers to work on projects in their area of responsibility.

The project tag includes the following child elements:
### Table 5–27  Project Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefCatalogs</td>
<td>References the subject areas that belong to the project. Each subject area is contained in a sub-element called RefPresentationCatalog.</td>
</tr>
<tr>
<td>RefFactTables</td>
<td>References the logical fact tables that belong to the project. Each fact table is contained in a sub-element called RefLogicalTable.</td>
</tr>
<tr>
<td>RefGroups</td>
<td>References the application roles that belong to the project. Each application role is contained in a sub-element called RefGroup.</td>
</tr>
<tr>
<td>RefUsers</td>
<td>References the users that belong to the project. Each user is contained in a sub-element called RefUser.</td>
</tr>
<tr>
<td>RefVariables</td>
<td>References the variables that belong to the project. Each variable is contained in a sub-element called RefVariable.</td>
</tr>
<tr>
<td>RefInitBlocks</td>
<td>References the initialization blocks that belong to the project. Each initialization block is contained in a sub-element called RefInitBlock.</td>
</tr>
<tr>
<td>RefListCatalogs</td>
<td>References the list catalogs that belong to the project. Each list catalog is contained in a sub-element called RefListCatalog.</td>
</tr>
<tr>
<td>RefTables</td>
<td>References the presentation tables that belong to the project. Each presentation table is contained in a sub-element called RefPresentationTable.</td>
</tr>
<tr>
<td>RefColumns</td>
<td>References the presentation columns that belong to the project. Each presentation column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>RefHierarchies</td>
<td>References the presentation hierarchies that belong to the project. Each presentation hierarchy is contained in a sub-element called RefPresentationHierarchy.</td>
</tr>
<tr>
<td>RefLevels</td>
<td>References the presentation levels that belong to the project. Each presentation level is contained in a sub-element called RefPresentationLevel.</td>
</tr>
<tr>
<td>RefExcludedCatalogs</td>
<td>References any subject areas that do not belong to the project, but that are mapped to logical objects that do belong to the project. Each subject area is contained in a sub-element called RefPresentationCatalog.</td>
</tr>
<tr>
<td>RefExcludedTables</td>
<td>References any presentation tables that do not belong to the project, but that are mapped to logical objects that do belong to the project. Each presentation table is contained in a sub-element called RefPresentationTable.</td>
</tr>
</tbody>
</table>
The project tag does not include any attributes.

**Syntax**

```xml
<Project _>
   <Description />
   <RefCatalogs>
      <RefPresentationCatalog catalogRef="" _/>
   </RefCatalogs>
   <RefFactTables>
      <RefLogicalTable logicalTableRef="" _/>
   </RefFactTables>
   <RefGroups>
      <RefGroup groupRef="" _/>
   </RefGroups>
   <RefUsers>
      <RefUser userRef="" _/>
   </RefUsers>
   <RefVariables>
      <RefVariable variableRef="" _/>
   </RefVariables>
   <RefInitBlocks>
      <RefInitBlock initBlockRef="" _/>
   </RefInitBlocks>
   <RefListCatalogs>
      <RefListCatalog listcatalogRef="" _/>
   </RefListCatalogs>
   <RefTables>
      <RefPresentationTable tableRef="" _/>
   </RefTables>
   <RefColumns>
      <RefPresentationColumn columnRef="" _/>
   </RefColumns>
   <RefHierarchies>
      <RefPresentationHierarchy hierarchyRef="" _/>
   </RefHierarchies>
   <RefLevels>
      <RefPresentationLevel levelRef="" _/>
   </RefLevels>
</Project _>
```

---

**Table 5–27 (Cont.) Project Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefExcludedColumns</td>
<td>References any presentation columns that do not belong to the project, but that are mapped to logical objects that do belong to the project. Each presentation column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>RefExcludedHierarchies</td>
<td>References any presentation hierarchies that do not belong to the project, but that are mapped to logical objects that do belong to the project. Each presentation hierarchy is contained in a sub-element called RefPresentationHierarchy.</td>
</tr>
<tr>
<td>RefExcludedLevels</td>
<td>References any presentation levels that do not belong to the project, but that are mapped to logical objects that do belong to the project. Each subject area is contained in a sub-element called RefPresentationLevel.</td>
</tr>
</tbody>
</table>
<RefExcludedCatalogs>
  <RefPresentationCatalog catalogRef="" .../>
</RefExcludedCatalogs>
<RefExcludedTables>
  <RefPresentationTable tableRef="" .../>
</RefExcludedTables>
<RefExcludedColumns>
  <RefPresentationColumn columnRef="" .../>
</RefExcludedColumns>
<RefExcludedHierarchies>
  <RefPresentationHierarchy hierarchyRef="" .../>
</RefExcludedHierarchies>
<RefExcludedLevels>
  <RefPresentationLevel levelRef="" .../>
</RefExcludedLevels>
</Project>

Example
<Project mdsid="m80c9e2bf-1075-0000-714b-e31d00000000"
  name="FUSION_Sale Forecasting"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.oracle.com/obis/repository">
  <RefCatalogs>
    <RefPresentationCatalog refId="m80c9e2bf-1075-0000-714b-e31d00000000-
      m80ca9dbb-0fa4-0000-714b-e31d00000000"
catalogRef="/oracle/bi/server/base/PresentationCatalog/Sales Manager
  Dashboard_u0150 Campaign 80ca9dbb-0fa4-0000-714b-e31d00000000.xml
  #m80ca9dbb-0fa4-0000-714b-e31d00000000" />
    <RefPresentationCatalog refId="m80c9e2bf-1075-0000-714b-e31d00000000-
      m80c9a197-0fa4-0000-714b-e31d00000000"
catalogRef="/oracle/bi/server/base/PresentationCatalog/Territory Management
  _u0150 CRM Forecast_80c9a197-0fa4-0000-714b-e31d00000000.xml#m80c9a197-
  0fa4-0000-714b-e31d00000000" />
    <RefPresentationCatalog refId="m80c9e2bf-1075-0000-714b-e31d00000000-
      m00224fba-07f3-0000-714b-e31d00000000"
catalogRef="/oracle/bi/server/base/PresentationCatalog/Sales Manager
  Dashboard_u0150 Campaign 80c9dbb-0fa4-0000-714b-e31d00000000.xml
  #m80c9a197-0fa4-0000-714b-e31d00000000" />
  </RefCatalogs>
  <RefFactTables>
    <RefLogicalTable refId="m80c9e2bf-1075-0000-714b-e31d00000000-m00224fba-07f3-
      0000-714b-e31d00000000"
  logicalTableRef="/oracle/bi/server/base/LogicalTable/Fact_u0150 CRM _u0150
  Revenue_00224fba-07f3-0000-714b-e31d00000000.xml#m00224fba-07f3-0000-714b-
  e31d00000000" />
    <RefLogicalTable refId="m80c9e2bf-1075-0000-714b-e31d00000000-m002250bc-07f3-
      0000-714b-e31d00000000"
  logicalTableRef="/oracle/bi/server/base/LogicalTable/Fact for Presentation
  Foldering_002250bc-07f3-0000-714b-e31d00000000.xml#m002250bc-07f3-0000-714b-
  e31d00000000" />
  </RefFactTables>
  <RefGroups>
    <RefGroup refId="m80c9e2bf-1075-0000-714b-e31d00000000-m0023e1a2-106e-0000-
      714b-e31d00000000"
  groupRef="/oracle/bi/server/base/Group/BIAAdministrators_0023e1a2-106e-0000-
  714b-e31d00000000.xml#m0023e1a2-106e-0000-714b-e31d00000000" />
  </RefGroups>
  <RefUsers>
    <RefUser refId="m80c9e2bf-1075-0000-714b-e31d00000000-m0021f847-05dd-0000-
      714b-e31d00000000"
  userRef="/oracle/bi/server/base/User/Administrator_0021f847-05dd-0000-714b-
  e31d00000000.xml#m0021f847-05dd-0000-714b-e31d00000000" />
  </RefUsers>
</Project>
5.3.2 MDS XML Elements: Query Object

The query object element corresponds to the saved query object in the repository metadata. You can save a query from the Query Repository dialog in the Administration Tool.

The query object tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryString</td>
<td>Contains the query string that qualifies and defines the query.</td>
</tr>
</tbody>
</table>

The query object tag does not include any attributes.

Syntax

```xml
<QueryObject ...>
  <Description />
  <QueryString .../>
</QueryObject>
```
Example

<QueryObject mdsid="mbd2048f1-0bf6-1000-806e-0a97078d0000"
    name="Sample Query"
    isDefaultReadable="false"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.oracle.com/obis/repository">
    <QueryString>
        <![CDATA[ (128 = ")" AND (6 = Database) AND (189 1 "1")]]>
    </QueryString>
</QueryObject>
Part II provides usage and reference information for the Oracle BI Server XML API.

Part II contains the following chapters:

- Chapter 6, "About the Oracle BI Server XML API"
- Chapter 7, "XML Logical Elements"
- Chapter 8, "XML Physical Elements"
- Chapter 9, "XML Presentation Elements"
- Chapter 10, "XML Marketing, Security, and Other Elements"
This chapter provides information about the Oracle BI Server XML API, which can be used to create a generic, XML-based representation of the Oracle BI repository metadata. The Oracle BI Server XML API can be used for metadata migration, programmatic metadata generation and manipulation, metadata patching, and other functions.

Note that Oracle BI Server XML API objects correspond to metadata repository objects in an RPD file. These objects are not the same as Oracle BI Presentation Catalog XML objects.

This chapter contains the following topics:

- Section 6.1, "About the Oracle BI Server XML API"
- Section 6.2, "Generating and Executing XML"
- Section 6.3, "Using the Oracle BI Server XML API"
- Section 6.4, "Generic XML Elements and Attributes"

6.1 About the Oracle BI Server XML API

The Oracle BI Server XML schema document (xudml1.xsd) contains the grammar for the structure of XML repository files. You can find this file at:

```
ORACLE_HOME/bifoundation/server/bin
```

Note that this is not the file that enables the generation of MDS XML output. For information about generating MDS XML output, see Chapter 1, "About the Oracle BI Server MDS XML API."

You can view this file to see relationships between XML metadata objects, but you must not modify this file. Doing so can lead to errors and crashes in the biserverxmlexec utility.

The structure of the repository is primarily a flat list of objects, in the format:

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<Repository xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DECLARE>
    <ObjectType (attributes)>
      Contents...
    </ObjectType>
    <ObjectType (attributes)>
      Contents...
    </ObjectType>
    ...
  </DECLARE>
</Repository>
```
Most of the modeling relationships, including the hierarchical ones, are formed using name references between objects, not by the hierarchical structure of the XML.

Note that the metadata objects in the XML file must appear in a particular order. You can view the set order in the XML schema document under the `objectList_t` element.

## 6.1.1 About Using the Oracle BI Server XML API to Merge and Append Objects

Do not use the Oracle BI Server XML API to attempt a full repository merge. The merge feature in the Administration Tool handles an extremely complex set of object comparisons and decisions, and it is not possible to account for and resolve these conflicts using the XML utilities. Because of this, using the XML API to perform a full merge is not supported. Attempting to do so might result in a corrupted repository.

However, you can use the XML API to append objects from one repository to another repository, using the `-B` option with `biserverxmlexec` to specify a base repository path name. This approach works when you have two completely separate repositories with no shared objects between the two, including no shared variables or application roles. For example, if you have separate repositories for the Credit and Manufacturing departments, you can take objects from the Credit XML metadata and append them to the Manufacturing XML metadata, and then convert to a single RPD that contains both working models. Using this method, you are essentially adding one repository to another.

Before attempting to perform an append operation, you should create a script to ensure that there are no common objects. Even when your repositories are separate, they might contain the same names for common objects, such as a `current_date` variable.

It is also a good idea to equalize the upgrade IDs to ensure that there are no accidental clashes. Equalizing objects causes objects in two repositories that have the same name but different upgrade ids to be treated as the same object. See "Equalizing Objects" in Oracle Fusion Middleware Metadata Repository Builder’s Guide for Oracle Business Intelligence Enterprise Edition for more information.

### 6.2 Generating and Executing XML

Oracle Business Intelligence provides the following command-line tools to generate and execute XML:

- **biserverxmlgen** generates XML from an existing RPD
- **biserverxmlexec** lets you execute the XML in offline mode to create or modify a repository file
- **biserverxmlcli** executes the XML against the Oracle BI Server

Before running the XML utilities, you must first run `bi-init` to launch a command prompt or shell window that is initialized to your Oracle instance. This utility is called `bi-init.sh` on Linux and UNIX systems, `bi-init.bat` on client installations of the Administration Tool, and `bi-init.cmd` for all other Windows installation types.

For example, on Windows:

1. In Windows Explorer, go to the location appropriate for your install type:
   - Client installations:
     ```
     ORACLE_HOME/bifoundation/server/bin
     ```
■ All other installations:

\[ ORACLE_INSTANCE/bifoundation/OracleBIApplication/coreapplication/setup \]

2. Double-click bi-init.cmd (or bi-init.bat for client installations) to display a command prompt that is initialized to your Oracle instance.

3. At the command prompt, type biserverxmlgen, biserverxmlexec, or biserverxmlcli with the desired options. You can also run the utilities with no arguments or parameters to see usage information.

Note that when you execute XML to update an existing repository with new content, objects are replaced in their entirety. For example, to change the name attribute of a PresentationTable object from name="Facts Revenue" to name="Facts Financial," you must submit the entire PresentationTable object to biserverxmlexec or biserverxmlcli.

In addition, make sure that the qualified name, UID, and ID are correct when replacing existing objects. The Oracle BI Server assumes that a different ID is a different object.

This section contains the following topics:

■ Section 6.2.1, "Generating XML from an Existing RPD"
■ Section 6.2.2, "Executing XML in Offline Mode"
■ Section 6.2.3, "Executing XML Against the Oracle BI Server"

6.2.1 Generating XML from an Existing RPD

Use biserverxmlgen to generate an XML file from an existing repository file (RPD). To use this utility, execute the following at the command prompt:

\[ biserverxmlgen -R repository_pathname [-P password] -O output_file_pathname [-8] [-N] [-Q|-S] \]

Where:

repository_pathname is the name and location of the repository file from which you want to generate XML.

password is the repository password.

The password argument is optional. If you do not provide a password argument, you are prompted to enter a password when you run the command. To minimize the risk of security breaches, Oracle recommends that you do not provide a password argument either on the command line or in scripts. Note that the password argument is supported for backward compatibility only, and will be removed in a future release.

For scripting purposes, you can pass the password through standard input.

output_file_pathname is the name and location of the XML output file you want to generate. It is recommended that you provide an output filename with an .xml extension.

Specify -8 to use UTF-8 encoding in the generated XML file. If you do not specify -8, the generated XML file uses ANSI ISO-8859-1 encoding by default.

Specify -N to generate output without upgrade ids (uids). These are the internal identifiers of objects. Use this if you want to create a new repository without the old upgrade ids.

You can use one of the two options -Q and -S to control the generation of security objects:
6.2.2 Executing XML in Offline Mode

Use biserverxmlexec to execute XML in offline mode. This utility can generate an entire repository file (RPD) from an XML repository file, or it can modify an existing repository file by applying the changes specified in the XML file. The input XML is validated against the Oracle BI Server XML schema. Any objects that exist in both the RPD and the XML file are overwritten by the corresponding object definitions in the XML file. Note that the same object cannot be defined twice in the XML file.

A qualified name (qualifiedName) or upgrade id (uid) must be specified for each object. When the utility resolves and identifies objects, uids have the highest priority, followed by qualifiedNames. See Section 6.4, “Generic XML Elements and Attributes” for more information about identifier attributes.

To use this utility, execute the following at the command prompt:

```
biserverxmlexec -I input_file_pathname [-B base_repository_pathname] [-P password] -O output_repository_pathname [-N]
```

Where:

- `input_file_pathname` is the name and location of the XML input file you want to execute.
- `base_repository_pathname` is the existing repository file you want to modify using the XML input file (optional). Do not specify this argument if you want to generate a new repository file from the XML input file.
- `password` is the repository password. If you specified a base repository, enter the repository password for the base repository. If you did not specify a base repository, enter the password you want to use for the new repository.
- The password argument is optional. If you do not provide a password argument, you are prompted to enter a password when you run the command. To minimize the risk of security breaches, Oracle recommends that you do not provide a password argument either on the command line or in scripts. Note that the password argument is supported for backward compatibility only, and will be removed in a future release. For scripting purposes, you can pass the password through standard input.
- `output_repository_pathname` is the name and location of the RPD output file you want to generate. Note that you must specify an output repository regardless of whether you are generating a new RPD, or whether you are modifying an existing base repository. The changes that result from executing the XML file on a base repository are contained in the specified output repository, not in the base repository.
- Specify `-N` to ignore all non-fatal errors. Examples of non-fatal errors are unresolved objects, duplicated objects, and broken expressions.

Examples

The following example creates a new repository called rp1:

```
biserverxmlexec -I testxudml.txt -O rp1.rpd
Give password: my_rpd_password
```
The following example modifies rp1 and writes to rp2:

```
biserverxmlexec -I testxudml.txt -B rp1.rpd -O rp2.rpd
Give password: my_rpd_password
```

### 6.2.3 Executing XML Against the Oracle BI Server

Use biserverxmlcli to execute XML against the Oracle BI Server. This utility is very similar to biserverxmlexec, but is instead used to modify an online repository. An online repository is a running Oracle BI Server instance that has loaded an existing repository into its navigation space.

To use this utility, execute the following at the command prompt:

```
biserverxmlcli -U user_name [-P user_password] [-R repository_password]
-D data_source_name [-w ssl_pass_phrase | -y] -I XML_file_pathname [-N]
```

Where:

- `user_name` is a valid user name. This user must have the manage repository permission (oracle.bi.server.manageRepositories).
- `user_password` is the corresponding user password. The user password argument is optional. If you do not provide a user password argument, you are prompted to enter a user password when you run the command. To minimize the risk of security breaches, Oracle recommends that you do not provide password arguments either on the command line or in scripts. Note that the user password argument is supported for backward compatibility only, and will be removed in a future release. For scripting purposes, you can pass the password through standard input.
- `repository_password` is the repository password for the online repository. The repository password argument is optional. If you do not provide a repository password argument, you are prompted to enter a repository password when you run the command. To minimize the risk of security breaches, Oracle recommends that you do not provide password arguments either on the command line or in scripts. Note that the repository password argument is supported for backward compatibility only, and will be removed in a future release. For scripting purposes, you can pass the password through standard input.
- `data_source_name` is the Oracle BI Server ODBC data source name (DSN) to which you want to connect.
- `ssl_pass_phrase` is the SSL passphrase for the Oracle BI Server, if the ODBC connection has been configured to use SSL.
- `-y` is the SSL passphrase to the private key file.
- `XML_file_pathname` is the name and location of the XML input file you want to execute.

Specify `-N` to ignore all non-fatal errors. Examples of non-fatal errors are unresolved objects, duplicated objects, and broken expressions.

### 6.3 Using the Oracle BI Server XML API

This section provides information about different use cases for the Oracle BI Server XML API feature. This section contains the following topics:

- Section 6.3.1, "Maintaining Versions of the Metadata"
6.3.1 Maintaining Versions of the Metadata

The Oracle BI Server XML API provides an efficient way of keeping versions of an RPD. Rather than storing entire RPD files for each version, you can generate XML patch files using the Create Patch feature in the Compare dialog in the Administration Tool. The patch file contains only the delta of the changes between the current and original versions. Using patch files to store changes eliminates the need to store redundant data in entire repository files and saves disk space. Note that you always need to retain a copy of the original repository file to maintain versions using this method.

You can apply a patch to the original file using the patch merge feature in the Administration Tool, using the patchrpd utility, or by running biserverxmlexec with the -B option.

For example, if you have repository versions A0, A1, and A2, and you have an XML patch file called A2-0delta.xml that contains the changes between the latest and original versions, you can run the following command:

```
biserverxmlexec -I A2-0delta.xml -B A0.rpd -O A2reproduced.rpd
Give password: my_password
```

Alternatively, if you stored changes between A0 and A1, and then A1 and A2, you can run two commands to apply the latest changes on the original file:

```
biserverxmlexec -I A1-0delta.xml -B A0.rpd -O A1reproduced.rpd
Give password: my_password
biserverxmlexec -I A2-1delta.xml -B A1reproduced.rpd -O A2reproduced.rpd
Give password: my_password
```

Make sure to specify different names for the -B and -O arguments to avoid overwriting the original repository file.


6.3.2 Migrating Metadata from Another System

If you have another business intelligence or reporting system and want to migrate the metadata to Oracle Business Intelligence, you can write an adapter to transform the third-party metadata to the Oracle Business Intelligence generic XML format. The XML schema file (xudml1.xsd) provides a way to understand the structure of the Oracle Business Intelligence metadata. After you have a compatible XML metadata file, you can use the biserverxmlexec utility to create a corresponding RPD file.
6.3.3 Applying a Metadata Patch Against a Running Oracle BI Server

If you use the Administration Tool to make changes to your repository in offline mode, you can use the XML patch feature to apply the changes to an online repository, without needing to restart the server or apply the changes manually.

To do this, open the new version of the repository in the Administration Tool, compare it with the running version, and create an XML patch file using the Create Patch feature. You can then apply the patch against the running Oracle BI Server to load the changes, using the biserverxmlcli utility. For example:

```
bserverxmlcli -U my_user -D my_server_dsn -I my_xml_file_pathname
Give user password: my_password
Give repository password: my_repos_password
```

6.3.4 Manipulating Metadata Programmatically

You can use the Oracle BI Server XML API to develop custom applications to modify Oracle BI repository metadata. To do this, you can generate XML from your RPD file using the biserverxmlgen utility, and then parse the XML using a generic XML parser. Because the Oracle BI XML utilities are command-line utilities, they can be invoked from batch files, scripts, or as processes from within the code. You can modify the XML programmatically, and then use the biserverxmlexec utility to generate a new RPD that includes the changes.

For example, the Cloning Application Framework (CAF) uses the Oracle BI Server XML API to replicate content developed for one application so that it can be used by another application. The goal of the CAF application is to reuse content across applications so that it does not need to be developed twice.

CAF users can map Presentation layer objects and logical dimension hierarchies from one RPD to the corresponding names in the other RPD. The CAF application generates XML for the first RPD using the biserverxmlgen utility, then scans the XML to determine the Presentation layer object dependencies in the Business Model and Mapping layer and Physical layer. It then creates the necessary Presentation and Business Model and Mapping layer objects (such as columns, subject areas, and expressions), identical to objects in the first RPD but with the appropriate mapped names. When the corresponding XML is ready, it generates the cloned RPD using the biserverxmlexec utility.

6.3.4.1 Example of Programmatically Creating a Repository Object Using XML

This example shows how to add a presentation column called "Billed Qty" to the "Facts Revenue" presentation table of the "Sample Small" subject area. The logical column in the business model on which it is based already exists, and has the name "Sample Small". "F0 Rev Base Measures". "2-01 Billed Qty (Sum All)".

Figure 6–1 illustrates the goal of the example.

![Figure 6–1 Logical Column Exists in Repository with No Corresponding Presentation Column](image)
To add the presentation column programmatically, two metadata changes are needed:

1. A `<PresentationColumn>` object must be created for the new "Billed Qty" presentation column. It must reference the "2-01 Bill Qty (Sum All)" logical column to which it maps, and the Facts Revenue presentation table to which it belongs.

2. The `<PresentationTable name="Facts Revenue">` object must be updated to reference the new "Billed Qty" presentation column object.

Figure 6–2 shows a view of what the new presentation column looks like in the Administration Tool after it is created.

![Administration Tool Equivalent for New Presentation Column](image)

To update the metadata, first create an XML file called Change.rpdx. This file will contain the metadata changes needed, in XML format. Add the following code:

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<Repository xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DECLARE>
    <PresentationColumn name="Billed Qty" parentName="&quot;Sample Small&quot;.. &quot;Facts Revenue&quot;" parentUid="4008:2143" parentUid="80cab573-07d0-0000-714b-e31d00000000" hasDispName="false" overrideLogicalName="true">
      <Description>
        <![CDATA[This presentation column was added using the XML metadata API.]]>
      </Description>
      <RefLogicalColumn uid="946" id="2006:106" qualifiedName="&quot;Sample Small&quot; &quot;F0 Rev Base Measures&quot; &quot;2-01 Billed Qty (Sum All)""/>
    </PresentationColumn>
  </DECLARE>
</Repository>
```

This code adds a presentation column that references both the parent presentation table to which it belongs ("Facts Revenue"), and the logical column to which it maps ("2-01 Billed Qty (Sum All)"). Note that although the new presentation column has a qualified name, the id and uid are not specified, because they will be automatically created by the Oracle BI Server when it loads the modified RPD file.
In addition to adding the presentation column object, you must also add a reference to the new object from the existing presentation table object. To do this, generate an XML version of the current repository using biserverxmlgen and search the XML for the name of the presentation table (Facts Revenue). Copy the code for the presentation table object to the clipboard, as follows:

```xml
<PresentationTable name="Facts Revenue" parentName="&quot;Sample Small&quot;.
&quot;&quot; parentId="4004:2137" parentUid="73609" id="4008:2143"
uid="73625"
hasDispName='false'>
  <Description>
    <![CDATA[This folder contains several types of aggregation metrics
    leveraging single revenue fact column. It also contains aggregations
    metrics based on Forecast physical table]]>
  </Description>
  <Aliases>
    <Alias name="F1 Revenue"/>
  </Aliases>
  <Columns>
    <RefPresentationColumn uid="80cbbe7b-07d6-0000-714b-e31d00000000"
      id="4010:1707" qualifiedName="&quot;Sample Small&quot;..
    Facts Revenue&quot;.&quot;Revenue&quot;/>
  </Columns>
</PresentationTable>
```

Paste the code for the Facts Revenue presentation table into the Change.rpdx file, and add the appropriate presentation column reference to it. The final Change.rpdx file appears as follows, with the key metadata changes shown in bold:

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<Repository xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DECLARE>
    <PresentationTable name="Facts Revenue" parentName="&quot;Sample Small&quot;.
      &quot;&quot; parentId="4004:2137" parentUid="80cab573-07d0-0000-714b-e31d00000000"
      id="4008:12457" uid="80cab839-0fa8-0000-714b-e31d00000000"
      hasDispName='false'>
      <Description>
        <![CDATA[This folder contains several types of aggregation metrics
        leveraging single revenue fact column. It also contains aggregations
        metrics based on Forecast physical table]]>
      </Description>
      <Aliases>
        <Alias name="F1 Revenue"/>
      </Aliases>
      <Columns>
        <RefPresentationColumn uid="80cb981e-0faa-0000-714b-e31d00000000"
          id="4010:1707" qualifiedName="&quot;Sample Small&quot;..
        Facts Revenue&quot;.&quot;Revenue&quot;/>
        <RefPresentationColumn qualifiedName="&quot;Sample Small&quot;..
          Facts Revenue&quot;.&quot;Billed Qty&quot;"/>
      </Columns>
    </PresentationTable>
    <PresentationColumn name="Billed Qty" parentName="&quot;Sample Small&quot;.
      ..&quot;Facts Revenue&quot;..&quot;Facts Revenue&quot;.&quot;Billed Qty&quot;"/>
    <RefPresentationColumn qualifiedName="&quot;Sample Small&quot;..
      Facts Revenue&quot;.&quot;Billed Qty&quot;"/>
      <Description>
        <![CDATA[This presentation column was added using the XML metadata API.]]>
      </Description>
  </DECLARE>
</Repository>
```
Finally, run the biserverxmlexec utility to execute Change.rpdx to update the original RPD file.

### 6.3.5 Moving from Test to Production Environments

When you move from test to production environments, you typically need to update connection pool settings for your data sources. You can use the Oracle BI Server XML API to programmatically update these settings in the repository.

To accomplish this task, you first need to open your test repository in the Administration Tool, manually update the connection pool settings for your data sources, and then save the repository under a different name. Next, you can compare the new repository with the original repository using the Administration Tool and then create an XML patch file that contains the changes. Finally, you can use biserverxmlexec or biserverxmlcli to execute the changes against a test repository.

Example 6–1 provides step-by-step information about how to perform these tasks using an example scenario.

See "Moving Oracle Business Intelligence to a Target Environment" in Oracle Fusion Middleware Administrator’s Guide for more information about moving from test to production environments in Oracle Business Intelligence.

---

**Example 6–1  Updating Connection Pool Test Values to Production Values**

This example shows how to programmatically update test values for two example connection pools to production values.

The following table shows example values for the two example data sources, in both test and production environments.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Data Source Type</th>
<th>Data Source Name</th>
<th>User Name</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>Relational</td>
<td>orcl_test</td>
<td>user_test</td>
<td>password_test</td>
</tr>
<tr>
<td>Test</td>
<td>XML</td>
<td><code>VALUEOF(ORACLE_HOME)/XML_data/Forecast</code></td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Production</td>
<td>Relational</td>
<td>orcl_prod</td>
<td>user_prod</td>
<td>password_prod</td>
</tr>
<tr>
<td>Production</td>
<td>XML</td>
<td><code>VALUEOF(ORACLE_HOME)/fcast_data/Forecast</code></td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

To programmatically update connection pool test values to production values:

1. Open the test repository (for example, SampleApp_test.rpd) in the Administration Tool.
2. In the Physical layer, open the connection pool for the relational source that needs to be changed.
3. Update the values for Data Source Name, User Name, and Password to the production values. Then, click OK.
4. Open the connection pool for the XML source that needs to be changed.

5. Update the value for **Data Source Name** to the production value. Then, click **OK**.

6. Select **File**, then select **Save As**. Then, save the repository under a new name (for example, **SampleApp_prod.rpd**).

7. Select **File**, then select **Compare**.

8. Select the original repository file (**SampleApp_test.rpd**) to compare to the currently open repository file (**SampleApp_prod.rpd**).

The Compare repositories dialog lists the two connection pools you modified

9. Click **Create Patch**, enter the name of the patch file you want to create (for example, **SampleApp_diff.xml**), and click **Save**.

The resulting XML patch file might appear as follows:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Repository xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DECLARE>
    <ConnectionPool name="SampleApp_XML" parentName='"&quot;1 - Sample App Xml Data&quot;' parentId='"3023:84' parentUid='"80ca62be-0bcf-0000-714b-e31d00000000' id='"3029:1158' uid='"80ca62c4-0bcf-0000-714b-e31d00000000' password='D7EDED84BC624A917F5B462A4DCA05CDE2566EEEEDC97D5AC4D07C3A079829F' timeout='300' maxConnDiff='10' maxConn='10' dataSource='VALUEOF(BI_EE_HOME)/fcast_data/Forecast' type='Default' reqQualifedTableName='false' isSharedLogin='false' isConcurrentQueriesInConnection='false' isCloseAfterEveryRequest='true' xmlRefreshInterval='4294967295' outputType='xml' bulkInsertBufferSize='0' transactionBoundary='0' xmlaUseSession='false' multiThreaded='false' supportParams='false' isSiebelJDBSecured='false'>
      <Description></Description>
    </ConnectionPool>
    <ConnectionPool name="SampleApp_Relational" parentName='"&quot;1 - Sample App Relational' parentId='"3023:85' parentUid='"80cc60c9-0bcf-0000-714b-e31d00000000' id='"3029:1159' uid='"80ca62bf-0bd5-0000-714b-e31d00000000' user='user_prod' password='D7EDED84BC624A917F5B462A4DCA05CDE2566EEEEDC97D5A1' timeout='4294967295' maxConnDiff='10' maxConn='10' dataSource='orcl_prod' type='OCI10G' reqQualifedTableName='false' isSharedLogin='true' isConcurrentQueriesInConnection='false' isCloseAfterEveryRequest='true' outputType='xml' bulkInsertBufferSize='131072' tempTablePrefix='TT' transactionBoundary='1000' xmlaUseSession='false' isSiebelJDBSecured='false'>
      <Description></Description>
    </ConnectionPool>
  </DECLARE>
</Repository>
```

10. After you create the patch, your test repository might continue to change, including name changes that could affect the parentNames or other attributes in your patch code. If this occurs, regenerate the patch using the steps in this section.

11. You can use biserverxmlexec to execute the generated XML patch file against the test version of the repository to programmatically update the connection pool settings. For example:

```bash
biserverxmlexec -I SampleApp_diff.xml -B SampleApp_test.rpd -O SampleApp_prod.rpd
```

Give password: my_rpd_password
To execute the XML patch file against a repository loaded into the Oracle BI Server, use biserverxmlcli.

6.3.6 Using XSL to Transform XML Files

You can use XSL to perform transformations on the XML files you generate. For example, you might want to use XSL to remove object IDs or UIDs to avoid conflicts.

Example 6–2 shows a sample XSL script that you can use as a model for your own transformations. The example script disables all logical table sources in the repository.

Example 6–2 Sample XSL Script

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
xmlns:xalan="http://xml.apache.org/xslt"
xmlns:fn="http://www.w3.org/2003/11/xpath-functions">
<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"
cdata-section-elements="Description Expr Value" xalan:indent-amount="4"/>
<xsl:strip-space elements="*"/>

<xsl:template match="/Repository/DECLARE">
  <xsl:element name="Repository">
    <xsl:element name="DECLARE">
      <xsl:for-each select="."/>
      <xsl:call-template name="objectmatch"/>
    </xsl:element>
  </xsl:element>
</xsl:template>

<!-- The 'objectmatch' template is matched for every metadata object -->
<xsl:template name="objectmatch">
  <xsl:choose>
    <!-- We copy every object except for Logical Table Sources. We call the 'ltsmatch' template for LTSs -->
    <xsl:when test="name() = 'LogicalTableSource'">
      <xsl:call-template name="ltsmatch"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:copy-of select="."/>
      <xsl:apply-templates/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- The 'ltsmatch' template gets matched for every logical table source. -->
<xsl:template name="ltsmatch">
  <xsl:element name="LogicalTableSource">
    <xsl:for-each select="@*"/>
    <xsl:call-template name="ltspropmatch"/>
    <xsl:for-each select="."/>
  </xsl:element>
</xsl:template>
```
6.3.7 Setting Permissions Using Command-Line Tools

You can set permissions for various presentation and marketing objects by using the command-line tools *biserverxmlexec* and *biserverxmlcli*.

To achieve the desired effect, first create a SetPermission element, together with the appropriate role, user, and object sub-elements in the XML file that is input to *biserverxmlexec* and *biserverxmlcli*. Then execute the command-line utility that you selected. The XML execution generates in the output repository the permissions that you specified in the SetPermission element.

For details, which include a SetPermission element example, see Section 10.2.11, "XML Elements: SetPermission."

6.4 Generic XML Elements and Attributes

This section describes generic elements and attributes that are common, like Description, name, and id. In the Oracle BI Server XML schema (.xsd file), these elements and attributes are defined under the *object_t* and *ref_t* elements.

The only common element is the Description element. This element can hold a text string describing a particular object. For example:

```xml
<Description>
  <! [CDATA[ Generated by Time Series Wizard ]]>
</Description>
```

There are eleven common attributes defined under the *object_t* element. Table 6–1 lists these attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>parentName</td>
<td>The qualified name of the parent of the object, in the format parent1.parent2.parent3, and so on.</td>
</tr>
<tr>
<td>parentId</td>
<td>The global unique identifier of the parent of the object, in the format type_id:parent_object_id.</td>
</tr>
</tbody>
</table>
In addition to the common attributes defined in object_t, additional common attributes are defined in ref_t. These attributes are used when one object references another object or set of objects. For example, business model objects reference the logical tables contained in that business model, using the sub-element RefLogicalTable. All referenced sub-elements have the same common attributes shown in Table 6–2.

### Table 6–2  Common XML Attributes in ref_t

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The global unique identifier of the object, in the format type_id:object_id.</td>
</tr>
<tr>
<td>uid</td>
<td>The internal identifier of the object. Referring to objects using this attribute instead of qualified names can provide faster access.</td>
</tr>
<tr>
<td>qualifiedName</td>
<td>The fully qualified name of the object. For example, the qualifiedName attribute for a logical column might appear as follows: <code>qualifiedName=&quot;&amp;quot;Sample Sales&amp;quot;.&amp;quot;F3 Rolling Measures&amp;quot;.&amp;quot;1-22L Revenue (All Hist Sum)&amp;quot;</code></td>
</tr>
</tbody>
</table>

**6.4.1 About Object Identifiers**

Oracle BI Server XML metadata objects can have three different types of object identifiers, any of which can be used to reference the object. Object identifiers include:

- **Fully qualified name (qualifiedName).** This identifier is the only reference you need to include in your XML when creating a new object, because the other identifiers (uid and id) are automatically generated by the Oracle BI Server when it opens the RPD file. The fully qualified name must be unique within the file, but the name (the text after the last dot) only needs to be unique within the scope of its parent.
The fully qualified name is the best identifier to use when searching an XML repository file to find a particular object, because the name of each object is visible in the Administration Tool. For example:

```
qualifiedName="&quot;Sample Sales&quot;.&quot;H0 Time&quot;.&quot;Month&quot;"
```

Note that qualified names for Presentation objects use "." between the subject area and the presentation table name. This syntax is used because there is no such thing as a presentation schema, which would normally go between a catalog and a table. For example:

```
qualifiedName="&quot;Sample Small&quot;.&quot;Facts Revenue&quot;.&quot;Billed Qty&quot;"
```

A similar convention is used for Physical layer objects, when the database does not have a physical catalog or physical schema.

- **uid.** This identifier, also called the UpgradeID, is a 32-bit number that is unique across the entire repository. The uid is created by the Oracle BI Server for the lifetime of the object. You can leave it blank when creating new objects in XML, because the server generates it automatically when it loads the RPD file. When you alter an object, make sure not to change the uid.

  The uid is used for compare and merge algorithms. It can also be useful for tracing relationships when analyzing or reverse-engineering XML metadata.

  For example: `uid="80ca62be-0bcf-0000-714b-e31d00000000"`

- **id.** This identifier, also called the VoidID, is created by the Oracle BI Server for the lifetime of the object. It is used in situations where the object type can be used to achieve better performance than the uid.

  Like the uid, you can leave the id blank when creating new objects in XML, because the server generates it automatically when it loads the RPD file. When you alter an object, make sure not to change the id.

  The id is a structure that contains two pieces of information: the type and a unique number within that type (for example, 2000:123456). Each object type, such as a logical column, has a particular number associated with it (for example, 2000). So, two objects of the same type both have the same number in the first part of the id.

  The unique number in the second part of the id is a randomly assigned number within that type. It must be unique within the type, but does not need to be unique across different object types. For example, a repository might have the valid ids 2000:123456 and 2001:123456.

  Although the id is typically created by the Oracle BI Server, you can create your own if needed, as long as you ensure that it is unique.

  For example: `id="4010:1720"`

### 6.4.2 About Naming Requirements for Repository Objects

All repository object names must follow these requirements:

- Names cannot be longer than 128 characters
- Names cannot contain leading or trailing spaces
- Names cannot contain single quotes, question marks, or asterisks

Note that repository object names can include multibyte characters.
This chapter provides Oracle BI Server XML API reference information for logical elements. Logical elements typically correspond to objects in the Business Model and Mapping layer of the Oracle BI repository, or in other words, the objects that appear in the center pane of the Oracle BI Administration Tool.

This chapter contains the following topics:

- Section 7.1, "XML Elements: Logical Column"
- Section 7.2, "XML Elements: Business Model"
- Section 7.3, "XML Elements: Logical Complex Join"
- Section 7.4, "XML Elements: Logical Level"
- Section 7.5, "XML Elements: Dimension"
- Section 7.6, "XML Elements: Logical Table"
- Section 7.7, "XML Elements: Logical Foreign Key"
- Section 7.8, "XML Elements: Logical Key"
- Section 7.9, "XML Elements: Logical Table Source"
- Section 7.10, "XML Elements: Logical Source Folder"
- Section 7.11, "XML Elements: Measure Definition"
- Section 7.12, "XML Elements: Aggregation Rule"
- Section 7.13, "XML Elements: Override Aggregation Rule"
- Section 7.14, "XML Elements: Logical Display Folder"

### 7.1 XML Elements: Logical Column

The logical column element corresponds to the logical column object in the Business Model and Mapping layer.

The logical column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDerived</td>
<td>When set to true, indicates that the logical column is based on a user-defined expression (for example, another logical column, a constant value, or a variable).</td>
</tr>
</tbody>
</table>
The logical column tag includes the following child elements:

### Table 7–2 Logical Column Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expr</strong></td>
<td>Contains the expression from which the logical column should be derived. This element is only used when the <code>isDerived</code> attribute is set to true.</td>
</tr>
<tr>
<td><strong>CustomSortOrder</strong></td>
<td>If the sort order for this logical column is based on a different logical column, this element references that logical column. The logical column upon which the sort order is based is contained in a sub-element called <code>RefLogicalColumn</code>.</td>
</tr>
<tr>
<td><strong>DescriptorID</strong></td>
<td>When multilingual columns are based on a lookup function, it is common to specify the non-translated lookup key column as the descriptor ID column of the translated column. This element references the descriptor ID column. The descriptor ID logical column is contained in a sub-element called <code>RefLogicalColumn</code>.</td>
</tr>
<tr>
<td><strong>Levels</strong></td>
<td>For dimension columns, shows the level to which this column has been assigned. For level-based measures, shows the level at which the column has been explicitly fixed. The level is contained in a sub-element called <code>RefLogicalLevel</code>.</td>
</tr>
</tbody>
</table>

### Syntax

```xml
<LogicalColumn isDerived="" isWriteable="">
  <Description />
  <Expr>...</Expr>
  <CustomSortOrder>
    <RefLogicalColumn/>
  </CustomSortOrder>
  <DescriptorID>
    <RefLogicalColumn/>
  </DescriptorID>
  <Levels>
    <RefLogicalLevel/>
  </Levels>
</LogicalColumn>
```

### Example

```xml
<LogicalColumn name="Person ID"
  parentName="&quot;8 - Fusion Order Demo (OLTP)&quot;.
  D01 Customers&quot;
  parentUid="80cbddc-07f3-0000-714b-e31d00000000"
  id="2006:61814" uid="80cbbe7b-07d6-0000-714b-e31d00000000"
  isWriteable="false">
  <Levels>
    <RefLogicalLevel/>
  </Levels>
</LogicalColumn>
```
7.2 XML Elements: Business Model

The business model element corresponds to the business model object in the Business Model and Mapping layer.

The business model tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isClassicStar</td>
<td>If set to true, indicates that the business model follows a star schema. Star schemas have one-to-many relationships between the logical dimension tables and the logical fact table.</td>
</tr>
<tr>
<td>isAvailable</td>
<td>If set to true, indicates that the corresponding subject area is available for queries.</td>
</tr>
</tbody>
</table>

The business model tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>References the display tables that belong to this business model. Each table is contained in a sub-element called RefLogicalTable. Included for backward compatibility. As table order is not significant within business model, this element will not be generated in output XML.</td>
</tr>
<tr>
<td>DisplayFolders</td>
<td>References the display folders that belong to this business model. Each display folder is contained in a sub-element called RefLogicalDisplayFolder.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<BusinessModel
    isClassicStar=""
    isAvailable="">
    <Description />
    <Tables>
        <RefLogicalTable />
        <RefLogicalTable />
    </Tables>
    <DisplayFolders>
        <RefLogicalDisplayFolder />
        <RefLogicalDisplayFolder />
    </DisplayFolders>
</BusinessModel>
```

Example

```xml
<BusinessModel name="9 - Usage Tracking"
7.3 XML Elements: Logical Complex Join

The logical complex join element corresponds to the logical join object in the Business Model and Mapping Layer.

The logical complex join tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of join. Valid values are Inner, LeftOuter, RightOuter, FullOuter, and FullOuterStitch.</td>
</tr>
</tbody>
</table>

The logical complex join tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Roles              | Roles define how a logical table or a logical level contributes to the join. Each role is contained in a sub-element called Role. The role sub-element has two attributes:  
  ■ multiplicity indicates the multiplicity of the table or level in the join. Valid values are $[(0-9)]^* \text{ TO } [(0-9)]^*$, ZERO TO ONE, ZERO TO MANY, ONE TO ONE, ONE TO MANY, and UNKNOWN.  
  ■ isAggregate indicates whether the table or level is an aggregate. Each Role sub-element can contain two additional sub-elements: RefLogicalTable, which references the logical tables in the join, and RefLogicalLevel, which references the logical levels in the join. See the syntax for the logical complex join element to see the complete hierarchy for the roles child element. |
| DrivingTable       | If a driving table has been specified for the key, this element references that logical table. The driving table is contained in a sub-element called RefLogicalTable. |

Syntax

```
<LogicalComplexJoin _
  type=""/>
<Description />
<Roles>
  <Role multiplicity="" isAggregate="">
    <RefLogicalTable _/>  
    <RefLogicalLevel _/>
  </Role>
</Roles>
<DrivingTable>
  <RefLogicalTable _/>
</DrivingTable>
```
Example

```xml
<LogicalComplexJoin name="Relationship_2004:617171273059824"
    id='2004:61717' uid='80cbbf09-07d4-0000-714b-e31d00000000'
    type='Inner'>
    <Roles>
        <Role multiplicity="n..n" isAggregate="false">
            <RefLogicalTable id='2035:61664'
                uid='80cbbdd9-07f3-0000-714b-e31d00000000'
                qualifiedName='"8 - Fusion Order Demo (OLTP)"'
                F1 Order Facts' />
        </Role>
        <Role multiplicity="0..1" isAggregate="false">
            <RefLogicalTable id='2035:61573'
                uid='80cbbdd5-07f3-0000-714b-e31d00000000'
                qualifiedName='"8 - Fusion Order Demo (OLTP)"'
                D11 Discounts and Coupons' />
        </Role>
    </Roles>
</LogicalComplexJoin>
```

7.4 XML Elements: Logical Level

The logical level element corresponds to the logical level object contained in the dimensions in the Business Model and Mapping layer.

The logical level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isUniqueSourceKey</td>
<td>If set to true, indicates that this level is associated with the logical column or columns that comprise the logical key of the dimension table (or in other words, that this level is the lowest level of the dimension).</td>
</tr>
</tbody>
</table>
| isGTA | If set to true, indicates that this level is the Grand Total level, which is a special level representing the grand total for a dimension. Each dimension can have just one Grand Total level. A grand total level does not contain dimensional attributes and does not have a level key. 
  **Note:** GTA stands for Grand Total Alias. |
| isFDUC | If set to true, indicates that the logical level rolls up to its parent. This option is called **Supports rollup to higher level of aggregation** in the Administration Tool. 
  **Note:** FDUC stands for FullDrillUpCoverage. |
| levelConst | The star schema level constant. |
| memberCount | The number of elements that exist at this logical level. |

The logical level tag includes the following child elements:
Table 7–8 Logical Level Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChildLevels</td>
<td>References the child levels that have been defined for this logical level. Each child level is contained in a sub-element called RefLogicalLevel.</td>
</tr>
<tr>
<td>PreferredDrillDown</td>
<td>If a drill path has been defined that is outside the normal drill path defined by the dimension level hierarchy, this element references the level to which users should drill. The preferred drill level is contained in a sub-element called RefLogicalLevel.</td>
</tr>
<tr>
<td>ParentKey</td>
<td>References the parent key for this level (for dimensions with parent-child hierarchies only). The parent key is contained in a sub-element called RefLogicalKey.</td>
</tr>
</tbody>
</table>

Syntax

```
<LogicalLevel ...
  isUniqueSourceKey=""
  isGTA=""
  isFDUC=""
  levelConst=""
  memberCount=""
  <Description />
  <ChildLevels>
    <RefLogicalLevel ...
    <RefLogicalLevel ...
  </ChildLevels>
  <PreferredDrillDown>
    <RefLogicalLevel ...
  </PreferredDrillDown>
  <ParentKey>
    <RefLogicalLevel ...
  </ParentKey>
</LogicalLevel>
```

Example

```
<LogicalLevel name="City"
  parentName="8 - Fusion Order Demo (OLTP)"
  H02 Cust Primary Addresses;
  parentUid="80cb715-07e3-0000-714b-e31d0000000"
  id="2025:63890" uid="80cbc796-07e9-0000-714b-e31d0000000"
  isGTA="false" isFDUC="true" memberCount="1" levelConst="City">
  <ChildLevels>
    <RefLogicalLevel id="2025:63894"
      uid="80cbc798-07e9-0000-714b-e31d0000000"
      qualifiedName="8 - Fusion Order Demo (OLTP)
      H02 Cust Primary Addresses;" />
  </ChildLevels>
</LogicalLevel>
```

7.5 XML Elements: Dimension

The dimension element corresponds to the dimension object in the Business Model and Mapping layer.
The dimension tag includes the following attributes:

**Table 7–9 Dimension Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isTimeDim</td>
<td>If set to true, indicates that this dimension is a time dimension.</td>
</tr>
<tr>
<td>isRagged</td>
<td>If set to true, indicates that this dimension hierarchy is unbalanced. An unbalanced hierarchy is one that contains levels that are not at the same depth.</td>
</tr>
<tr>
<td>isSkipped</td>
<td>If set to true, indicates that this dimension hierarchy is a skip-level hierarchy. A skip-level hierarchy is one in which there are members that do not have a value for a particular parent level.</td>
</tr>
<tr>
<td>isValueBased</td>
<td>If set to true, indicates that this object is a dimension with parent-child hierarchies. A parent-child hierarchy is a hierarchy of members that all have the same type. In a parent-child hierarchy, the inter-member relationships are parent-child relationships between dimension members.</td>
</tr>
</tbody>
</table>

The dimension tag includes the following child elements:

**Table 7–10 Dimension Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RootLevels</td>
<td>References the root level or levels of this dimension. Root levels are levels in the hierarchy that do not have any levels above them. In the Administration Tool, root levels appear directly below the dimension icon. Each level is contained in a sub-element called RefLogicalLevel.</td>
</tr>
<tr>
<td>DefaultRootLevel</td>
<td>References the default root level of this dimension. When there are many root levels in a dimension, the default root level is the one that is used for drilldown. The default root level is contained in a sub-element called RefLogicalLevel.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<Dimension...
  isTimeDim=""
  isRagged=""
  isSkipped=""
  isValueBased="">
  <Description />
  <RootLevels>
    <RefLogicalLevel .../>
    <RefLogicalLevel .../>
  </RootLevels>
  <DefaultRootLevel>
    <RefLogicalLevel .../>
  </DefaultRootLevel>
</Dimension>
```

**Example**

```xml
<Dimension name="H0 Time"
  parentName="&quot;3 - Sample Federated&quot;"
  parentId="2000:53458" parentUid="80cb9462-07d0-0000-714b-e31d00000000"...
```
7.6 XML Elements: Logical Table

The logical table element corresponds to the logical table object in the Business Model and Mapping layer.

The logical table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isLookupTable</td>
<td>If set to true, indicates that this logical table is a lookup table. Lookup tables are typically used to store fields that contain translations in several languages. Queries that require multilingual data join the base table and lookup table to obtain the translated values for each row in the base table.</td>
</tr>
<tr>
<td>x</td>
<td>The x coordinate of the table when it is shown in the business model diagrams.</td>
</tr>
<tr>
<td>y</td>
<td>The y coordinate of the table when it is shown in the business model diagrams.</td>
</tr>
</tbody>
</table>

The logical table tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LevelColumn</td>
<td>References the column that is used in the level that corresponds to this logical table. Each level column is contained in a sub-element called RefLogicalColumn.</td>
</tr>
<tr>
<td>Columns</td>
<td>References the logical columns that belong to this logical table. Each logical column is contained in a sub-element called RefLogicalColumn.</td>
</tr>
<tr>
<td>TableSources</td>
<td>References the logical table sources for this logical table. Each logical table source is contained in a sub-element called RefLogicalTableSource.</td>
</tr>
</tbody>
</table>

**Syntax**

```
<LogicalTable ...
    isLookupTable='' />
```
7.7 XML Elements: Logical Foreign Key

The logical foreign key element corresponds to the logical foreign key object in the Business Model and Mapping layer.

The logical foreign key tag includes the following attributes:
The logical foreign key tag includes the following child elements:

**Table 7–13 Logical Foreign Key Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of join. Valid values are Inner, LeftOuter, RightOuter, FullOuter, and FullOuterStitch.</td>
</tr>
</tbody>
</table>

**Table 7–14 Logical Foreign Key Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>References the logical columns in the table that the foreign key references. Each logical column is contained in a sub-element called RefLogicalColumn.</td>
</tr>
<tr>
<td>CounterPartKey</td>
<td>References the foreign key columns for this foreign key. Each logical column is contained in a sub-element called RefLogicalKey.</td>
</tr>
<tr>
<td>DrivingTable</td>
<td>If a driving table has been specified for this logical foreign key, this element references that logical table. The driving table is contained in a sub-element called RefLogicalTable.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<LogicalForeignKey _
   type=""/>
   <Description />  
   <Columns>
     <RefLogicalColumn _/>
     <RefLogicalColumn _/>
   </Columns>
   <CounterPartKey>
     <RefLogicalKey _/>
   </CounterPartKey>
   <DrivingTable>
     <RefLogicalTable _/>
   </DrivingTable>
</LogicalForeignKey>
```

**Example**

```xml
<LogicalForeignKey name='Measures_FKey'
   parentName='9 - Usage Tracking; &quot;Measures&quot;'
   parentID='2035:22319' parentUid='802b28a4-07f3-0000-714b-e31d00000000'
   id='2033:22559' uid='802b29a5-07f1-0000-714b-e31d00000000'
   type='Inner'>
   <Columns>
     <RefLogicalColumn id='2006:22349'
       uid='802b28b2-07d6-0000-714b-e31d00000000'
       qualifiedName='9 - Usage Tracking; &quot;Measures&quot; QueryText'>
   </Columns>
   <CounterPartKey>
     <RefLogicalKey id='2008:22412'
       uid='802b29a0-07d8-0000-714b-e31d00000000'
       qualifiedName='9 - Usage Tracking; &quot;Topic&quot; Topic_Key'>
   </CounterPartKey>
</LogicalForeignKey>
```
7.8 XML Elements: Logical Key

The logical key element corresponds to the logical key object in the Business Model and Mapping layer.

The logical key tag includes the following attributes:

**Table 7–15  Logical Key Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isPrimary</td>
<td>If set to true, indicates that this key is the logical primary key for the table.</td>
</tr>
<tr>
<td>isForDrillDown</td>
<td>If set to true, indicates that this key is the key that is displayed when a user clicks to drill down.</td>
</tr>
<tr>
<td>isChronKey</td>
<td>If set to true, indicates that this key is a chronological key for a time dimension.</td>
</tr>
</tbody>
</table>

The logical key tag includes the following child elements:

**Table 7–16  Logical Key Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>References the logical column that defines the key of the logical table. The logical column is contained in a sub-element called RefLogicalColumn.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<LogicalKey ...
  isPrimary=""
  isForDrillDown=""
  isChronKey="">
  <Description />
  <Columns>
    <RefLogicalColumn .../>
  </Columns>
</LogicalKey>
```

**Example**

```xml
<LogicalKey name="CUSTOMER_ID"
  parentName="&quot;8 - Fusion Order Demo (OLTP)&quot;.
  F2 Coupon Usages&quot;" parentUid="80cbbe8-07f3-0000-714b-e31d00000000"
  parentId="2035:61501" parentUid="80cbbde8-07f3-0000-714b-e31d00000000"
  id="2008:61513" uid="80cbbf39-07d8-0000-714b-e31d00000000"
  isPrimary="true">
  <Columns>
    <RefLogicalColumn id="2006:61512"
      uid="80cbbf04-07d6-0000-714b-e31d00000000"
      qualifiedName="&quot;8 - Fusion Order Demo (OLTP)&quot;.
      F2 Coupon Usages&quot; Order ID&quot;" />
  </Columns>
</LogicalKey>
```
7.9 XML Elements: Logical Table Source

The logical table source element corresponds to the logical table source object in the Business Model and Mapping layer. See also the column mapping element, described in Section 7.9.1, "Column Mapping."

The logical table source tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDistinct</td>
<td>This attribute is used if the values for the logical table source are unique. In the Administration Tool, this option is called Select distinct values in the Content tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>isDimOnly</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>isSysGen</td>
<td>If set to true, indicates that an internal utility such as the Aggregate Persistence Wizard has generated this logical table source.</td>
</tr>
<tr>
<td>canCombine</td>
<td>If set to true, indicates that this logical table source can be combined with other logical table sources to provide a full set of results. In the Administration Tool, this option is called This source should be combined with other sources at this level in the Content tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>isActive</td>
<td>If set to true, indicates that this logical table source is active. In the Administration Tool, this attribute corresponds with the Disabled option in the General tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>allowUnmappedTable</td>
<td>Determines if the logical table source can have physical tables which are not mapped to logical columns. This can occur with a snowflake of physical tables A-&gt;B-&gt;C, where a logical table maps to columns in A and C, but B needs to be included in the logical table source because it is in the join path between A and C.</td>
</tr>
<tr>
<td>isJoinRestricted</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>priorityGroup</td>
<td>Indicates the priority group number of this logical table source. Logical table source priority group numbers are assigned to indicate which logical table source should be used for queries for which there is more than one logical table source that can satisfy the requested set of columns.</td>
</tr>
</tbody>
</table>

The logical table source tag includes the following child elements:
Table 7–18  Logical Table Source Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>References the set of physical tables to which the logical table source is mapped. These links are references to the physical joins that already exist. In the Administration Tool, this element corresponds with the Map to these tables section in the General tab of the Logical Table Source dialog. Each link child element contains two sub-elements, StartNode and Joins:</td>
</tr>
<tr>
<td></td>
<td>■ The StartNode sub-element references the physical table identified by this link. It can contain the RefCubeTable, RefPhysicalTable, RefAWDimension, or RefAWCube, depending on your data source type.</td>
</tr>
<tr>
<td></td>
<td>■ The Joins sub-element defines the joins for this link. It contains different sub-elements, depending on the type of join. For example:</td>
</tr>
<tr>
<td></td>
<td>RefCubeTable, RefPhysicalTable, RefAWDimension, or RefAWCube</td>
</tr>
<tr>
<td></td>
<td>RefPhysicalForeignKey or RefComplexJoin</td>
</tr>
<tr>
<td></td>
<td>DrivingTable, which contains a sub-element RefPhysicalTable</td>
</tr>
<tr>
<td></td>
<td>The Joins sub-element has two attributes: type (Inner, LeftOuter, RightOuter, FullOuter, or FullOuterStitch) and cardinality (ONE TO ONE, ZERO OR ONE TO ONE, ONE TO ZERO OR ONE, ZERO OR ONE TO ZERO OR ONE, ONE TO MANY, ZERO OR ONE TO MANY, ZERO OR ONE TO MANY, MANY TO ONE, MANY TO ZERO OR ONE, or MANY TO MANY). See the syntax for the logical table source element to see the complete hierarchy for the link child element.</td>
</tr>
<tr>
<td>WhereClause</td>
<td>If a WHERE clause filter has been defined, contains the WHERE clause expression that limits the number of rows the source uses in the resultant table. The expression is contained in a sub-element called Expr.</td>
</tr>
<tr>
<td></td>
<td>In the Administration Tool, this option is called Use this &quot;WHERE clause&quot; filter to limit rows returned (exclude the &quot;WHERE&quot;) in the Content tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>GroupBy</td>
<td>Contains the expression that defines whether the aggregation is grouped by logical level or by column. The expression is contained in a sub-element called Expr.</td>
</tr>
<tr>
<td></td>
<td>In the Administration Tool, this option is called Aggregation content, group by in the Content tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>FragmentContent</td>
<td>Contains the expression that defines how the source is fragmented. The expression is contained in a sub-element called Expr.</td>
</tr>
<tr>
<td></td>
<td>This child element is used in conjunction with the canCombine attribute. In the Administration Tool, this option is called Fragmentation content in the Content tab of the Logical Table Source dialog.</td>
</tr>
<tr>
<td>ClosureTable</td>
<td>References the parent-child table, if any. The parent-child table is contained in a sub-element called RefPhysicalTable.</td>
</tr>
</tbody>
</table>
**Syntax**

```xml
<LogicalTableSource
    isDistinct=""
    isSysGen=""
    canCombine=""
    isActive=""
    allowUnmappedTable=""
    priorityGroup=""/>
<Description />
<Link>
    <StartNode>
        <RefPhysicalTable />
    </StartNode>
    <Joins>
        <Join type="" cardinality="">
            <RefPhysicalTable />
            <RefPhysicalTable />
            <RefPhysicalForeignKey />
        </Join>
    </Joins>
    <WhereClause>
        <Expr>...</Expr>
    </WhereClause>
    <groupBy>
        <Expr />
    </groupBy>
    <FragmentContent>
        <Expr />
    </FragmentContent>
    <ClosureTable>
        <RefPhysicalTable />
    </ClosureTable>
</LogicalTableSource>
```

**Example**

```xml
<LogicalTableSource name="LTS3 Channels (Bridged to Revenue)"
    parentName="1 - Sample App&D4 Offices and Channels (M:N Joins)" parentId="2035:11640" parentUid="80cab595-07f3-0000-714b-e31d00000000"
    parentId="2037:12357" uid="80cab7d5-07f5-0000-714b-e31d00000000"
    isActive="true">
    <Link>
        <StartNode>
            <RefPhysicalTable id="3001:42412"
                uid="80cb6825-0bb9-0000-714b-e31d00000000"
                qualifiedName="01 - Sample App (ORCL).Catalog.BISAMPLE.D30 Offices"/>
        </StartNode>
        <Joins>
            <Join type="Inner" cardinality="ONE TO MANY">
                <RefPhysicalTable id="3001:42412"
                    uid="80cb6825-0bb9-0000-714b-e31d00000000"
                    qualifiedName="01 - Sample App Data (ORCL)&quot;Catalog&amp;quot;&amp;quot;BISAMPLE&amp;quot;&amp;quot;D30 Offices&amp;quot;"/>
                <RefPhysicalTable id="3001:42416"
                    uid="80cb6829-0bb9-0000-714b-e31d00000000"
                    qualifiedName="01 - Sample App Data (ORCL)&quot;Catalog&amp;quot;&amp;quot;BISAMPLE&amp;quot;&amp;quot;D43 Chnls Offs Bridge (Rev)"/>
            </Join>
        </Joins>
    </Link>
</LogicalTableSource>
```
**7.9.1 Column Mapping**

The column mapping element does not correspond to any metadata repository object. It is only used in conjunction with the logical table source element. The column mapping element appears on its own in the XML schema (.xsd file), rather than as part of the logical table source element, so that the column mapping details are easier to read and update.

The column mapping tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefLogicalTableSource</td>
<td>References the logical table source to which this column mapping belongs. Each column mapping must specify one and only one logical table source.</td>
</tr>
<tr>
<td>LogicalColumn</td>
<td>Contains the expression that identifies the logical column for this column mapping. The expression is contained in a sub-element called Expr.</td>
</tr>
<tr>
<td>Expr</td>
<td>Contains the expression that identifies the physical column for this column mapping. Note that this Expr child element is different from the Expr sub-element that appears inside the LogicalColumn child element.</td>
</tr>
</tbody>
</table>

The column mapping tag does not include any attributes.

**Syntax**

```xml
<ColumnMapping>
  <RefLogicalTableSource _/>  
  <LogicalColumn>
    <Expr>...</Expr>
  </LogicalColumn>
</ColumnMapping>
```
7.10 XML Elements: Logical Source Folder

The logical source folder element corresponds to the logical source folder object in the Business Model and Mapping layer. Logical source folders contain logical table sources.

The logical source folder element does not include any attributes nor child elements, other than the default attributes and elements.

Syntax

```xml
<LogicalSourceFolder ...
  <Description />
</LogicalSourceFolder>
```

Example

```xml
<LogicalSourceFolder name="Sources"
  parentName="Usage Tracking" parentUid="6c898eac-b03c-e91a-14ac-7868e974aefc"
  parentId="2035:43831" id="2046:43884" uid="6c898eac-b03c-e91a-14ac-7868e974aefc">
  
</LogicalSourceFolder>
```

7.11 XML Elements: Measure Definition

The measure definition element stores the measure definition for a measure column. A measure column is usually in a fact table, can change for each record, and can be added up or aggregated in some way.

The measure definition tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isCommutative</td>
<td>If set to true, indicates that the aggregation rule lets the server switch the order if it can increase the performance. For example, the following aggregation function is commutative: $A \ (B (x)) = B \ (A (x))$</td>
</tr>
</tbody>
</table>
The measure definition tag includes the following child elements:

Table 7–21 Measure Definition Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules</td>
<td>References the aggregation rule for this measure. Each rule is contained in a sub-element called RefAggrRule. The order of rules is important. For more information, see Section 7.12, &quot;XML Elements: Aggregation Rule.&quot;</td>
</tr>
<tr>
<td>OverrideRules</td>
<td>References the override aggregation rule for this measure, if any. Each override aggregation rule is contained in a sub-element called RefOverrideAggrRule. The order of override rules is important. For more information, see Section 7.13, &quot;XML Elements: Override Aggregation Rule.&quot;</td>
</tr>
</tbody>
</table>

Syntax

```
<MeasureDefn _
   isCommutative=''
   <Rules />
   <OverrideRules />
   <Description />
</MeasureDefn>
```

Example

```
<MeasureDefn name="Measure"
   parentName="&quot;SampleApp Lite&quot;.&quot;F0 Revenue Base Measures&quot;.
   &quot;Billed Quantity&quot;'
   parentId="2006:42434"
   parentUid="80cb683b-07d6-0000-714b-e31d00000000"
   id="2048:42595" uid="80cb68dc-0800-0000-714b-e31d00000000"
   isCommutative='false'>
   <Rules>
       <RefAggrRule id="2050:43457" uid="80cb6c51-0802-0000-714b-e31d00000000"
                qualifiedName="SampleApp Lite'.'F0 Revenue Base Measures'."Billed Quantity".
                'Measure'.'AggRule_2050:434571273647807" />
   </Rules>
</MeasureDefn>
```

7.12 XML Elements: Aggregation Rule

The aggregation rule element corresponds to the aggregation rule object in the Business Model and Mapping layer.

The aggregation rule tag includes the following attributes:

Table 7–22 Aggregation Rule Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDefault</td>
<td>If set to true, indicates that this is the default rule for aggregate calculations. This option only applies when there is more than one aggregation rule.</td>
</tr>
</tbody>
</table>

The aggregation rule tag includes the following child elements:
### 7.13 XML Elements: Override Aggregation Rule

The override aggregation rule element corresponds to the override aggregation expression for a logical table source in the Business Model and Mapping Layer.

When you select Count Distinct as the default aggregation rule, you can specify an override aggregation expression for specific logical table sources. This option is useful when you have more than one logical table source mapped to a logical column and you want to apply a different aggregation rule to each source.

The override aggregation rule tag includes the following child elements:

| **Table 7–24 Override Aggregation Rule Child Elements** |
|-----------------------------------------------|-------------------------------------------------|
| **Element Name** | **Description** |
| RefLogicalTableSource | References the logical table source to which this override aggregation rule applies. |
| Expr | Contains the expression on which this aggregation rule is based. |

The override aggregation rule tag does not include any attributes.

**Syntax**

```xml
<OverrideAggrRule _>
</OverrideAggrRule>
```
7.14 XML Elements: Logical Display Folder

The logical display folder element corresponds to the logical display folder object in the Business Model and Mapping layer.

The logical display folder tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>References the dimensions that belong to this logical display folder. Each dimension is contained in a sub-element called RefDimension.</td>
</tr>
<tr>
<td>Tables</td>
<td>References the logical tables that belong to this logical display folder. Each table is contained in a sub-element called RefLogicalTable.</td>
</tr>
</tbody>
</table>

The logical display folder element does not include any attributes.

Syntax

```xml
<LogicalDisplayFolder …>
  <Description />
  <Dimensions>
    <RefDimension …/>
    <RefDimension …/>
  </Dimensions>
  <Tables>
    <RefLogicalTable …/>
    <RefLogicalTable …/>
  </Tables>
</LogicalDisplayFolder>
```
Example

<LogicalDisplayFolder name="Resources"
    parentName="SampleApp"
    parentId="2000:42377" parentUid="80cb6802-07d0-0000-714b-e31d00000000"
    id="2052:44085" uid="9bf9c250-0c00-1000-835d-0a9702490000">
    <Tables>
        <RefLogicalTable id="2035:42395"
            uid="80cb6814-07f3-0000-714b-e31d00000000"
            qualifiedName="SampleApp.D0 Time"/>
        <RefLogicalTable id="2035:42562"
            uid="80cb68bb-07f3-0000-714b-e31d00000000"
            qualifiedName="SampleApp.D2 Offices"/>
    </Tables>
</LogicalDisplayFolder>
This chapter provides Oracle BI Server XML API reference information for physical elements. Physical elements typically correspond to objects in the Physical layer of the Oracle BI repository, or in other words, the objects that appear in the right-hand pane of the Oracle BI Administration Tool.

This chapter contains the following topics:

- Section 8.1, "XML Elements: Database"
- Section 8.2, "XML Elements: Physical Catalog"
- Section 8.3, "XML Elements: Physical Schema"
- Section 8.4, "XML Elements: Physical Display Folder"
- Section 8.5, "XML Elements: Physical Table"
- Section 8.6, "XML Elements: Physical Column"
- Section 8.7, "XML Elements: Connection Pool"
- Section 8.8, "XML Elements: Physical Key"
- Section 8.9, "XML Elements: Physical Foreign Key"
- Section 8.10, "XML Elements: Complex Join"
- Section 8.11, "XML Elements: Cube Table"
- Section 8.12, "XML Elements: Physical Dimension"
- Section 8.13, "XML Elements: Physical Hierarchy"
- Section 8.14, "XML Elements: Physical Level"
- Section 8.15, "XML Elements: Cube Column"
- Section 8.16, "XML Elements: Analytic Workspace"
- Section 8.17, "XML Elements: AW Cube"
- Section 8.18, "XML Elements: AW Dimension"
- Section 8.19, "XML Elements: AW Hierarchy"
- Section 8.20, "XML Elements: AW Level"
- Section 8.21, "XML Elements: AW Column"

8.1 XML Elements: Database

The database element corresponds to the database object in the Physical layer.
The database tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of data source, such as XMLDirect or ODBC20.</td>
</tr>
<tr>
<td>dbFlag</td>
<td>Stores the value of the following options:</td>
</tr>
<tr>
<td></td>
<td>- <strong>CRM metadata tables</strong>: For legacy Siebel Systems sources only. When selected, the Administration Tool import utility looks for the table definition in Oracle’s Siebel CRM-specific tables.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Virtual Private Database</strong>: Identifies the physical database source as a virtual private database (VPD). When a VPD is used, returned data results are contingent on the user’s authorization credentials. This option is used with the Security Sensitive option for session variables.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Allow populate queries by default</strong>: When selected, allows all users to execute <code>POPULATE SQL</code>. If you want most, but not all, users to be able to execute <code>POPULATE SQL</code>, select this option and then limit queries for specific users or groups.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Allow direct database requests by default</strong>: When selected, allows all users to execute physical queries. The Oracle BI Server will send unprocessed, user-entered, physical SQL directly to an underlying database. If you want most, but not all, users to be able to execute physical queries, select this option and then limit queries for specific users or groups.</td>
</tr>
</tbody>
</table>

The database tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>Lists the SQL features for this database. Each feature is contained in a sub-element called <code>Feature</code>. This sub-element has two attributes:</td>
</tr>
<tr>
<td></td>
<td>- <strong>name</strong>: indicates the name of the feature, such as <code>LEFT_OUTER_JOIN_SUPPORTED</code>.</td>
</tr>
<tr>
<td></td>
<td>- <strong>value</strong>: indicates whether this feature is supported by the database, or provides the actual value of the feature (such as &quot;0&quot; for <code>MAX_COLUMNS_IN_SELECT</code>).</td>
</tr>
<tr>
<td>PersistConnPool</td>
<td>References the connection pool that is used as the persist connection pool, if one has been assigned. A persist connection pool is a database property that is used for specific types of queries (typically used to support Marketing queries). The persist connection pool is contained in a sub-element called <code>RefConnectionPool</code>.</td>
</tr>
<tr>
<td>DisplayFolders</td>
<td>References any physical display folders that contain objects in this database. Physical display folders can be used to organize tables in a database. For example, all alias tables can be stored in a display folder called Alias, in addition to being present in the database. Each display folder is contained in a sub-element called <code>RefPhysicalDisplayFolder</code>.</td>
</tr>
</tbody>
</table>
Table 8–2 (Cont.) Database Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionPools</td>
<td>References the connection pools for this database object. Each connection pool is contained in a sub-element called RefConnectionPool.</td>
</tr>
<tr>
<td>Properties</td>
<td>Custom database properties for Oracle ADF Business Component data sources that are passed to the Administration Tool from Oracle ADF BI view objects.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<Database _
    type=""
    dbFlag=''>
    <Description />
    <Features>
        <Feature name='' value=''/>
        <Feature name='' value=''/>
        ...
    </Features>
    <PersistConnPool>
        <RefConnectionPool _/>
    </PersistConnPool>
    <DisplayFolders>
        <RefPhysicalDisplayFolder _/>
        <RefPhysicalDisplayFolder _/>
    </DisplayFolders>
    <ConnectionPools>
        <RefConnectionPool _/>
    </ConnectionPools>
    <Properties>
        <PropertyGroup>
            category=""
            <Property>
                <Name _/>
                <Value _/>
            </Property>
            ...
        </PropertyGroup>
    </Properties>
</Database>
```

Example

```xml
<Database name="Paint" id="3023:205" uid="80ca62be-0bcf-0000-714b-e31d00000000" type="XMLDirect">
    <Features>
        <Feature name="LEFT_OUTER_JOIN_SUPPORTED" value="false"/>
        <Feature name="RIGHT_OUTER_JOIN_SUPPORTED" value="false"/>
        <Feature name="FULL_OUTER_JOIN_SUPPORTED" value="false"/>
        <Feature name="NESTED_OUTER_JOIN_SUPPORTED" value="false"/>
        <Feature name="UNION_SUPPORTED" value="false"/>
        <Feature name="UNION_ALL_SUPPORTED" value="false"/>
        <Feature name="COUNT_SUPPORTED" value="false"/>
        <Feature name="COUNT_DISTINCT_SUPPORTED" value="false"/>
        ...
        <Feature name="SORT_ORDER_LOCALE" value="english-usa"/>
        <Feature name="COMMENT_START"/>
        <Feature name="COMMENT_END"/>
    </Features>
</Database>
```
8.2 XML Elements: Physical Catalog

The physical catalog element corresponds to the physical catalog object in the Physical layer.

The physical catalog tag includes the following child elements:

<table>
<thead>
<tr>
<th>Table 8–3</th>
<th>Physical Catalog Child Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Name</td>
<td>Description</td>
</tr>
<tr>
<td>DynamicName</td>
<td>If a session variable is being used to specify the name of this physical catalog, this element references that session variable. The session variable is contained in a sub-element called RefVariable.</td>
</tr>
</tbody>
</table>

The physical catalog element does not include any attributes.

Syntax

```xml
<PhysicalCatalog ...
  <Description />
  <DynamicName>
    <RefVariable .../>
  </DynamicName>
</PhysicalCatalog>
```

Example

```xml
<PhysicalCatalog name="Catalog"
  parentName="01 - Sample App Data (ORCL)"
  parentId="3023:85" parentUid="80ca62c4-0bcf-0000-714b-e31d00000000"
  id="3025:42377" uid="80cb6802-0bd1-0000-714b-e31d00000000">
</PhysicalCatalog>
```

8.3 XML Elements: Physical Schema

The physical schema element corresponds to the physical schema object in the Physical layer.

The physical schema tag includes the following child elements:

<table>
<thead>
<tr>
<th>Table 8–4</th>
<th>Physical Schema Child Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Name</td>
<td>Description</td>
</tr>
<tr>
<td>DynamicName</td>
<td>If a session variable is being used to specify the name of this physical schema, this element references that session variable. The session variable is contained in a sub-element called RefVariable.</td>
</tr>
<tr>
<td>InputMessageTemplate</td>
<td>This child element is reserved for a future release.</td>
</tr>
</tbody>
</table>

The physical schema element does not include any attributes.

Syntax

```xml
<Schema ...
  <Description />
```
8.4 XML Elements: Physical Display Folder

The physical display folder element corresponds to the physical display folder object in the Physical layer.

The physical display folder tag includes the following child elements:

<table>
<thead>
<tr>
<th>Table 8–5 Physical Display Folder Child Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element Name</strong></td>
</tr>
<tr>
<td>Tables</td>
</tr>
</tbody>
</table>

The physical display folder element does not include any attributes.

Syntax

```xml
<PhysicalDisplayFolder ...>
  <Description />  
  <Tables>
    <RefPhysicalTable .../>
  </Tables>
</PhysicalDisplayFolder>
```

Example

```xml
<PhysicalDisplayFolder name="Assets" parentName="Sample Data" parentUid="80ca6538-0bb9-0000-714b-e31d00000000"
  qualifiedName="&quot;Sample Data&quot;...&quot;D20 Offices&quot;"/>
</PhysicalDisplayFolder>
```

8.5 XML Elements: Physical Table

The physical table element corresponds to the physical table object in the Physical layer.

The physical table tag includes the following attributes:
### Table 8–6 **Physical Table Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| type | The type of physical table. Options are as follows:  
- `select` indicates that this physical table object is a `SELECT` statement.  
- `sp` indicates that this physical table is a stored procedure.  
- `alias` indicates that this physical table is an alias table. An alias table (alias) is a physical table that references a different physical table as its source.  
- `none` indicates that this physical table is a regular physical table. |
| isCacheable | If set to true, indicates that this table is included in the Oracle BI Server query cache. |
| CacheExpiry | How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire. |
| isCacheEventTable | If set to true, indicates that this table has been registered as an Oracle BI Server event polling table. |
| isOverrideCacheProp | For alias tables only. If set to true, indicates that the alias table has its own cache properties that override the cache properties of the source table. |
| pollFreq | The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds. |
| x | The x coordinate of the table when it is shown in the physical diagram. |
| y | The y coordinate of the table when it is shown in the physical diagram. |
| rowCount | The latest number of rows retrieved for this physical table. |
| lastUpdated | The time at which row counts were last updated for this physical table. |
| uri | For XML data sources only. The resource identifier for the XML data source. |
| xsltPath | For XML data sources only. An XSLT file contains formatting rules written according to the XSLT standard. It defines how an XML file may be transformed. |
| maxConn | For XML data sources only. The maximum number of connections allowed to the same URL from different users. |
| xmlRefreshInterval | For XML data sources only. The time interval after which the XML data source is queried again directly, rather than using results in cache. |
| scriptPath | For XML data sources only. The extended gateway script path. |
| hints | For Oracle Databases only. Lists any database hints, such as index hints or leading hints, that were specified for this physical table. |
| isExtGatewayInProcess | For XML data sources only. Indicates that this physical table is the extended gateway part of the process. |
| extGatewayDelimiter | For XML data sources only. The row delimiter used in the metadata import process, typically the second level element in the XML document. |
The physical table tag includes the following child elements:

**Table 8–7  Physical Table Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceTable</td>
<td>For alias tables only. References the physical table that is being used as the source for this alias table. The physical table is contained in a sub-element called RefPhysicalTable.</td>
</tr>
<tr>
<td>DBMap</td>
<td>Stores the value of database-specific strings. The <code>&lt;DBMap&gt;</code> element contains an <code>&lt;Item&gt;</code> tag, which in turn contains a <code>&lt;Value&gt;</code> tag, as follows: <code>&lt;Item name=&quot;&quot;&gt; &lt;Value /&gt; &lt;/Item&gt;</code></td>
</tr>
<tr>
<td>AggrConnPool</td>
<td>If this physical table is an aggregate table, this element references the connection pool used for the aggregate. The connection pool is contained in a sub-element called RefConnectionPool.</td>
</tr>
<tr>
<td>RefVariable</td>
<td>If a session variable is being used to specify the name of this physical table, this element references that session variable.</td>
</tr>
<tr>
<td>MemberKey</td>
<td>Applies to parent-child relationship tables that support parent-child hierarchies on relational sources. The name of the column that uniquely identifies members of the parent-child relationship table. The member key is contained in a sub-element called RefPhysicalColumn.</td>
</tr>
<tr>
<td>ParentKey</td>
<td>The name of the column that contains the key of the parent or ancestor member. Applies to parent-child relationship tables that support parent-child hierarchies on relational sources. The parent key is contained in a sub-element called RefPhysicalColumn.</td>
</tr>
<tr>
<td>DistanceColumn</td>
<td>The name of the column that records the number of levels between the member and the ancestor. Applies to parent-child relationship tables that support parent-child hierarchies on relational sources. The distance column is contained in a sub-element called RefPhysicalColumn.</td>
</tr>
</tbody>
</table>
Table 8–7 (Cont.) Physical Table Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeafColumn</td>
<td>The name of the column that indicates whether the member has no child members (1 if the member is a leaf and has no children, 0 if the member is not a leaf and has children). Applies to parent-child relationship tables that support parent-child hierarchies on relational sources. The leaf column is contained in a sub-element called RefPhysicalColumn.</td>
</tr>
<tr>
<td>XSLT</td>
<td>For Oracle web service data sources. XSLT stores the XSLT transformation to flatten the web service response into a tabular result set.</td>
</tr>
<tr>
<td>XSDSchema</td>
<td>For Oracle web service data sources. XSDSchema stores the XML schema describing the structure of the web service response.</td>
</tr>
<tr>
<td>Properties</td>
<td>Contains one or more Property sub-elements, each of which has attributes &lt;name&gt; and &lt;value&gt;. These attributes specify a name-value pair of properties for data source metadata. Used for ADF sources, to store information related to the ADF application, such as information on flex Vos, override data types, join definitions, and data filters.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<PhysicalTable ...
    type=""
    isCacheable=""
    cacheExpriy=""
    isCacheEventTable=""
    isOverrideCacheProp=""
    pollFreq=""
    x=""
    y=""
    rowCount=""
    lastUpdated=""
    uri=""
    xslPath=""
    maxConn=""
    xmlRefreshInterval=""
    scriptPath=""
    hints=""
    isExtGatewayInProcess=""
    extGatewayDelimiter=""
    isSysGen=""
    isDDLAggr=""
    isDeployedView=""/>
<Description />
<SourceTable>
  <RefPhysicalTable ...
</SourceTable>
</DBMap>
<Item name="">
  <Value ...
</Item>
</DBMap>
<AggrConnPool>
  <RefConnectionPool ...
</AggrConnPool>
```
8.6 XML Elements: Physical Column

The physical column element corresponds to the physical column object in the Physical layer.

The physical column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The external name for the physical column. This attribute is required if the same name (such as STATE) is used in multiple hierarchies. This attribute is optional for XML documents. For XML, the extName attribute stores the fully qualified name of a column.</td>
</tr>
</tbody>
</table>
The physical column tag includes the following child elements:

### Table 8–9 Physical Column Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynamicName</td>
<td>If a session variable is being used to specify the name of this physical column, this element references that session variable. The session variable is contained in a sub-element called RefVariable.</td>
</tr>
<tr>
<td>SourceColumn</td>
<td>For physical columns in an alias table only. References the physical column that is being used as the source for this column. The physical column is contained in a sub-element called RefPhysicalColumn.</td>
</tr>
<tr>
<td>PicklistColumns</td>
<td>A list of physical columns to which the picklist is mapped. Each physical column is contained in a sub-element called RefPhysicalColumn.</td>
</tr>
<tr>
<td>Properties</td>
<td>Contains one or more Property sub-elements, each of which has attributes &lt;name&gt; and &lt;value&gt;. These attributes specify a name-value pair of properties for data source metadata.</td>
</tr>
</tbody>
</table>
Syntax

```xml
<PhysicalColumn _
    extName=""
dataType=""
    length=""
    nullable=""
    precision=""
    scale=""
    rowCount=""
    lastUpdated=""
    valueType=""
    mappedColumn=""
    MVLinkName=""
    specialType="">
    <Description />
    <DynamicName>
        <RefVariable _/>
    </DynamicName>
    <SourceColumn>
        <RefPhysicalColumn _/>
    </SourceColumn>
    <PicklistColumns>
        <RefPhysicalColumn _/>
    </PicklistColumns>
    <Properties>
        <Property name="" value=""/>
        <Property name="" value=""/>
    </Properties>
</PhysicalColumn>
```

Example

```xml
<PhysicalColumn name="Prod_Dsc"
    parentName="\"01 - Sample App Data (ORCL)\".Catalog\".BISAMPLE\".SAMP_PRODUCTS_D\"
    parentId="3001:42392" parentUid="80cb6811-0bb9-0000-714b-e31d00000000"
    id="3003:42538" uid="80cb68a3-0bbb-0000-714b-e31d00000000"
    dataType="VARCHAR"
    precision="25"
    rowCount="0.0"
    specialType="none">
    <Description>
        <![CDATA[Product Description: Stores the product description information as text]]>
    </Description>
</PhysicalColumn>
```

8.7 XML Elements: Connection Pool

The connection pool element corresponds to the connection pool object in the Physical layer.

The connection pool tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bulkInsertBufferSize</td>
<td>Used for limiting the number of bytes each time data is inserted in a database table. The value is given in kilobytes (KB).</td>
</tr>
</tbody>
</table>
### Table 8–10 (Cont.) Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transactionBoundary</td>
<td>Controls the batch size for an insert in a database table.</td>
</tr>
<tr>
<td>unicodeDB</td>
<td>This attribute is typically set to true when working with columns of an explicit Unicode data type, such as NCHAR, in an Unicode database.</td>
</tr>
<tr>
<td>tempTablePrefix</td>
<td>The first two characters in a temporary table name. The default value is TT.</td>
</tr>
<tr>
<td>tempTableOwner</td>
<td>Table owner name used to qualify a temporary table name in a SQL statement, for example to create the table owner.tablename.</td>
</tr>
<tr>
<td>tempTableDatabase</td>
<td>Database where the temporary table will be created. This property applies only to IBM OS/390, because IBM OS/390 requires the database name qualifier to be part of the CREATE TABLE statement.</td>
</tr>
<tr>
<td>tempTableSpace</td>
<td>Tablespace where the temporary table will be created. This property applies only to IBM OS/390, because IBM OS/390 requires the tablespace name qualifier to be part of the CREATE TABLE statement.</td>
</tr>
<tr>
<td>user</td>
<td>The user name for the data source.</td>
</tr>
<tr>
<td>password</td>
<td>The password for the data source. Note that passwords are encrypted.</td>
</tr>
<tr>
<td>timeout</td>
<td>The amount of time that a connection to the data source will remain open after a request completes. During this time, new requests use this connection rather than open a new one (up to the number specified for the maximum connections). The time is reset after each completed connection request. If you set the timeout to 0 (the default), connection pooling is disabled. In other words, each connection to the data source terminates immediately when the request completes. Any new connections either use another connection pool or open a new connection.</td>
</tr>
<tr>
<td>maxConnDiff</td>
<td>The maximum number of total connections allowed to this connection pool across all users.</td>
</tr>
<tr>
<td>dataSource</td>
<td>The data source name that is configured to access the database to which you want to connect. The data source name must contain valid logon information for the data source.</td>
</tr>
<tr>
<td>reqQualifiedTableName</td>
<td>If set to true, indicates that this database requires fully qualified table names. The fully qualified names are based on the physical object names in the repository.</td>
</tr>
<tr>
<td>isSharedLogin</td>
<td>If set to true, all users whose queries use the connection pool will access the underlying database using the same user name and password.</td>
</tr>
<tr>
<td>isConcurrentQueriesInConnection</td>
<td>An option that enables a single database connection to remain open for the specified time for use by future query requests. If you do not select this option, each query sent to the database opens a new connection. In the Administration Tool, this option corresponds to a value of “selected” for Enable connection pooling in the General tab of the Connection Pool dialog.</td>
</tr>
</tbody>
</table>
Table 8–10 (Cont.) Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isCloseAfterEveryRequest</td>
<td>If set to true, indicates that the connection pool gets closed after every query. In the Administration Tool, this option corresponds to a value of &quot;not selected&quot; for Enable connection pooling in the General tab of the Connection Pool dialog.</td>
</tr>
<tr>
<td>isAsyncExecution</td>
<td>An option that indicates that the data source supports asynchronous queries.</td>
</tr>
<tr>
<td>executeOnCmd</td>
<td>A command that is to be executed by the Oracle BI Server each time a connection is made to the database. The command may be any command accepted by the database.</td>
</tr>
<tr>
<td>xsiPath</td>
<td>For XML data sources only. An XSLT file contains formatting rules written according to the XSLT standard. It defines how an XML file may be transformed. Specifying the XSLT file in the connection pool applies it to all the XML physical tables in the connection pool. Used when outputType is set to XML.</td>
</tr>
<tr>
<td>maxConn</td>
<td>The maximum number of total connections allowed to this connection pool for a given user.</td>
</tr>
<tr>
<td>xmlRefreshInterval</td>
<td>For XML data sources only. The time interval after which the XML data source is queried again directly, rather than using results in cache.</td>
</tr>
<tr>
<td>scriptPath</td>
<td>The connection method Search script. This option is used for XML Server data sources.</td>
</tr>
<tr>
<td>outputType</td>
<td>For XML data sources only. The query output format (delimited, xml, or fixed). In the Administration Tool, this option is called Query output format in the XML tab of the Connection Pool dialog. If this option is not set, the query output format is assumed to be a custom format, and the value for convertScriptPath is used.</td>
</tr>
<tr>
<td>gwDelim</td>
<td>The delimiting characters for the query output format. Used when outputType is set to delimited.</td>
</tr>
<tr>
<td>xpath</td>
<td>An XPath expression is a simple XSLT transformation rule that fits into one line. Used when outputType is set to xml.</td>
</tr>
<tr>
<td>convertScriptPath</td>
<td>Used when outputType is not set. The path to the .exe file to be used to convert the query output to a custom output format (such as HTML).</td>
</tr>
<tr>
<td>ignoreFirstLine</td>
<td>If set to true, indicates that the first line of the output file is ignored.</td>
</tr>
<tr>
<td>xmlHeaderPath</td>
<td>The path to the header file, if any, used as a query input supplement. Used for XML Server data sources.</td>
</tr>
<tr>
<td>trailerPath</td>
<td>The path to the trailer file, if any, used as a query input supplement. Used for XML Server data sources.</td>
</tr>
<tr>
<td>supportParams</td>
<td>Indicates whether the data source supports parameters. If set to false, and the database features table supports parameters, special code executes that enables the Oracle BI Server to push filters (or calculations) with parameters to the database.</td>
</tr>
</tbody>
</table>
XML Elements: Connection Pool

### Table 8–10 (Cont.) Connection Pool Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| isolationLevel | For ODBC and DB2 gateways. Sets the transaction isolation level on each connection to the back-end database. The isolation level setting controls the default transaction locking behavior for all statements issued by a connection. Options are as follows:  
  - default uses the default transaction locking behavior of the data source.  
  - readCommitted specifies that shared locks are held while the data is read to avoid dirty reads.  
  - readUncommitted implements dirty read (isolation level 0 locking). When this option is set, it is possible to read uncommitted or dirty data, change values in the data, and have rows appear or disappear in the data set before the end of the transaction.  
  - repeatableRead places locks on all data that is used in a query, preventing other users from updating the data.  
  - serializable places a range lock on the data set, preventing other users from updating or inserting rows into the data set until the transaction is complete. |
| xmlaInfo | The vendor-specific information used to connect to the multidimensional data source. |
| xmlaCatalog | The list of catalogs available. The cube tables correspond to the catalog you use in the connection pool. |
| xmlaUseSession | If set to true, causes queries go through a common session. |
| multiThreaded | If set to true, indicates that the connection pool supports multi-threading, or in other words, that one connection pool can support multiple queries. If this option is set to false, each query is tied to a single database connection. |
| type | The call interface type. |
| appServerName | The name of the ADF application server. |
| appServerPort | The port number of the ADF application server. |
| appID | The ADF application ID. |
| appEJBBean | The name of the ADF application Enterprise Java Bean. |
| javaHostServiceName | The name of the JavaHost service. The JavaHost is used as a connector between the Oracle BI Server and data sources that use Java, such as ADF. |
| isSiebelJDBSecuredisaw | For Siebel OM data sources only. Indicates whether the connection between the Oracle BI Server and the Siebel data source is encrypted. |
| csfKey | This attribute is not currently used. |
| isAW | This attribute is not currently used. |
| useCert | This attribute stands for “Use Certificate.” It is used for ADF data sources and indicates whether to use a certificate when securing the connection. |
| wsServer | Specifies the Oracle database web service host name. |
| wsWSDL | Specifies the WSDL URL to describe the Oracle database web service. |

The connection pool tag includes the following child elements:
Table 8–11  Connection Pool Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AfterQueryString</td>
<td>Contains a connection script that is executed after the query is run.</td>
</tr>
<tr>
<td>BeforeQueryString</td>
<td>Contains a connection script that is executed before the query is run.</td>
</tr>
<tr>
<td>PostConnectString</td>
<td>Contains SQL queries executed after the connection is closed.</td>
</tr>
<tr>
<td>PreDisconnectString</td>
<td>Contains SQL queries executed before the connection is established.</td>
</tr>
<tr>
<td>RelationalConnPool</td>
<td>This child element is not used.</td>
</tr>
<tr>
<td>RelationalDataFolder</td>
<td>This child element is not used.</td>
</tr>
<tr>
<td>SQLBypassDB</td>
<td>For Oracle ADF data sources only. References the SQL Bypass database, if one has been specified. The SQL Bypass feature directly queries the database so that aggregations and other transformations are pushed down where possible, reducing the amount of data streamed and worked on in Oracle Business Intelligence. The database object is contained in a sub-element called RefDatabase.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<ConnectionPool _
  bulkInsertBufferSize="" 
  transactionBoundary="" 
  unicodeDB="" 
  tempTablePrefix="" 
  tempTableOwner="" 
  tempTableDatabase="" 
  tempTableSpace="" 
  user="" 
  password="" 
  timeout="" 
  maxConnDiff="" 
  dataSource="" 
  reqQualifiedTableName="" 
  isSharedLogin="" 
  isConcurrentQueriesInConnection="" 
  isCloseAfterEveryRequest="" 
  isAsyncExecution="" 
  executeOnCmd="" 
  xslPath="" 
  maxConn="" 
  xmlRefreshInterval="" 
  scriptPath="" 
  outputType="" 
  gwDelim="" 
  xpath="" 
  convertScriptPath="" 
  ignoreFirstLine="" 
  xmlHeaderPath="" 
  trailerPath="" 
  supportParams="" 
  isolationLevel="" 
  xmlaInfo="" 
  xmlaCatalog="" 
```
8.8 XML Elements: Physical Key

The physical key element corresponds to the physical key object in the Physical layer. The physical key tag includes the following child elements:
The physical key element does not include any attributes.

**Syntax**

```xml
<PhysicalKey _>
  <Description />
  <Columns>
    <RefPhysicalColumn _/>
    <RefCubeColumn _/>
  </Columns>
</PhysicalKey>
```

**Example**

```xml
<PhysicalKey name="09 Segment_Key"
  parentName="01 - Sample App Data (ORCL)".Catalog".
  .BISAMPLE".D61 Customer Segments"
  parentId="3001:42429" parentUid="80cb6836-0bb9-0000-714b-e31d00000000"
  id="3008:43143" uid="80cb6b00-0bc0-0000-714b-e31d00000000">
  <Columns>
    <RefPhysicalColumn id="3003:42857"
      uid="80cb69e2-0bbb-0000-714b-e31d00000000"
      qualifiedName="01 - Sample App Data (ORCL)".Catalog".
      .BISAMPLE".D61 Customer Segments"
      .Segment_Key"
  </Columns>
</PhysicalKey>
```

### 8.9 XML Elements: Physical Foreign Key

The physical foreign key element corresponds to the physical foreign key object in the Physical layer.

The physical foreign key tag includes the following attributes:

**Table 8–13  Physical Foreign Key Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hints</td>
<td>For Oracle Databases only. Lists any database hints, such as index hints or leading hints, that were specified for this physical foreign key.</td>
</tr>
</tbody>
</table>

The physical foreign key tag includes the following child elements:

**Table 8–14  Physical Foreign Key Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>References the physical columns in the table that the foreign key references. Each physical column is contained in a sub-element called either RefPhysicalColumn or RefCubeColumn.</td>
</tr>
</tbody>
</table>
8.10 XML Elements: Complex Join

The complex join element corresponds to the complex join object in the Physical layer. The complex join tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hints</td>
<td>For Oracle Databases only. Lists any database hints, such as index hints or leading hints, that were specified for this complex join.</td>
</tr>
<tr>
<td>cardinality</td>
<td>Indicates the cardinality of the join. Valid values are: ONE TO ONE, ZERO OR ONE TO ONE, ONE TO ZERO OR ONE, ZERO OR ZERO OR ONE, ONE TO MANY, ZERO OR ONE TO MANY, ZERO OR ONE TO MANY, MANY TO ONE, MANY TO ZERO OR ONE, or MANY TO MANY.</td>
</tr>
</tbody>
</table>
The complex join tag includes the following child elements:

<table>
<thead>
<tr>
<th>Table 8–16  Complex Join Child Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element Name</strong></td>
</tr>
<tr>
<td>Tables</td>
</tr>
<tr>
<td>Expr</td>
</tr>
<tr>
<td>ExternalExpr</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<ComplexJoin _
    hints="" cardinity="" extName="">
    <Description />
    <Tables>
        <RefPhysicalTable _/>
        <RefPhysicalTable _/>
    </Tables>
    <Expr>  <![CDATA["01 - Sample App Data (ORCL)"."Catalog"."BISAMPLE"."D04 Time Rolling Mth"."Mth_Key" >= "01 - Sample App Data (ORCL)"."Catalog"."BISAMPLE"."F90 Facts Rolling Time Mth"."Mth_Key"]]> </Expr>
    <ExternalExpr>  <![CDATA["01 - Sample App Data (ORCL)"."Catalog"."BISAMPLE"."F90 Facts Rolling Time Mth"."Mth_Key"]]> </ExternalExpr>
</ComplexJoin>
```

**Example**

```xml
<ComplexJoin name="D04 Time Rolling Mth_F90 Facts Rolling Time Mth"
    id="3012:43263" uid="80cb6b78-0bc4-0000-714b-e31d00000000">
    <RefPhysicalTable id="3001:42446"
        uid="80cb6b78-0bc4-0000-714b-e31d00000000"
        qualifiedName="&quot;01 - Sample App Data (ORCL)&quot;.&quot;Catalog&quot;.
        &quot;BISAMPLE&quot;.&quot;D04 Time Rolling Mth&quot; />
    <RefPhysicalTable id="3001:42407"
        uid="80cb6b78-0bc4-0000-714b-e31d00000000"
        qualifiedName="&quot;01 - Sample App Data (ORCL)&quot;.&quot;Catalog&quot;.
        &quot;BISAMPLE&quot;.&quot;D04 Time Rolling Mth&quot; />
    <Expr>  <![CDATA["01 - Sample App Data (ORCL)"."Catalog"."BISAMPLE"."D04 Time Rolling Mth"."Mth_Key" >= "01 - Sample App Data (ORCL)"."Catalog"."BISAMPLE"."F90 Facts Rolling Time Mth"."Mth_Key"]]> </Expr>
</ComplexJoin>
```

### 8.11 XML Elements: Cube Table

The cube table element corresponds to the physical cube table object in the Physical layer.
The cube table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The external name for the data source.</td>
</tr>
</tbody>
</table>
| type | The type of physical cube table. Options are as follows:  
  - `select` indicates that this physical table object is a `SELECT` statement.  
  - `sp` indicates that this physical cube table is a stored procedure.  
  - `alias` indicates that this physical cube table is an alias table to a logical table source.  
  - `none` indicates that this physical cube table is a regular physical cube table. |
| isCacheable | If set to true, indicates that this table is included in the Oracle BI Server query cache. |
| CacheExpiry | How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire. |
| isCacheEventTable | If set to true, indicates that this table has been registered as an Oracle BI Server event polling table. |
| pollFreq | The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds. |
| x | The x coordinate of the table when it is shown in the physical diagram. |
| y | The y coordinate of the table when it is shown in the physical diagram. |
| rowCount | The latest number of rows retrieved for this physical cube table. |
| lastUpdated | The time at which row counts were last updated for this physical cube table. |
| scriptPath | For XML data sources only. The extended gateway script path. |
| isSysGen | If set to true, indicates that an internal utility has generated this physical cube table. |
| isOverrideCacheProp | For alias tables only. If set to true, indicates that the alias table has its own cache properties that override the cache properties of the source table. |
| hints | This attribute is not used. |
| isDDLAggr | Indicates whether this cube table is an aggregate table. |
| memberReference | For Essbase data sources only. Indicates the cube property (for example, Member Name). An Essbase cube can have multiple aliases, and the value of memberReference is one of the alias names. The query engine will generate a query against the alias set for memberReference. |
| dateFormat | For Essbase only. Defines the format of date member, for example, 'yyyy-MM-dd.' The value is automatically set during import. |
| isMemberAliasUnique | This is used for query performance. If you know that all the member names in your data source are unique, you can set it to true by selecting the option **Use unqualified member name for better performance.** |
The cube table tag includes the following child elements:

**Table 8–18 Cube Table Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>For Essbase data sources only. For any alias tables for this Essbase cube, lists the alias to use (such as default, LongNames, French, and so on).</td>
</tr>
<tr>
<td>AliasTableVar</td>
<td>For Essbase data sources only. Each AliasTableVar represents an Essbase alias table. The AliasTableVar child element contains a sub-element called RefVariable.</td>
</tr>
<tr>
<td>DBMap</td>
<td>This child element is not used.</td>
</tr>
</tbody>
</table>
| CubeVars       | For SAP/BW data sources only. Cube variables are used as a means of parameterizing queries. Each cube variable is contained in a sub-element called CubeVar. This sub-element has the following attributes:  
  ■ name: Name of the cube variable.  
  ■ caption: A description (label or caption) associated with the cube variable, mainly used for display purposes.  
  ■ type: The type of cube variable (SAP_VAR_TYPE_MEMBER, SAP_VAR_TYPE_HIERARCHY, or SAP_VAR_TYPE_NUMERIC).  
  ■ selectionType: The selection type of the cube variable, for cube variables of type SAP_VAR_TYPE_MEMBER (SAP_VAR_SEL_TYPE_VALUE, SAP_SEL_TYPE_INTERVAL, or SAP_VAR_SEL_TYPE_COMPLEX).  
  ■ entryType: Indicates whether replacing variables is optional or mandatory. Values include SAP_VAR_INPUT_TYPE_OPTIONAL, SAP_VAR_INPUT_TYPE_MANDATORY, or SAP_VAR_INPUT_TYPE_MANDATORY_NOT_INITIAL.  
  ■ refDimension: The reference dimension. This column contains a DIMENSION_UNIQUE_NAME for the parameter type SAP_VAR_TYPE_HIERARCHY.  
  ■ refHierarchy: The reference hierarchy. This column contains a HIERARCHY_UNIQUE_NAME for the variable type SAP_VAR_TYPE_MEMBER.  
  ■ defaultLow: Contains a default value for the variable, or is zero.  
  ■ defaultHigh: Contains a default value for the variable, or is zero. This property is only important for variables with the selection type SAP_VAR_SEL_TYPE_INTERVAL and SAP_VAR_SEL_TYPE_SELECTION.  
  ■ overrideLow: The default value for the cube variable (used if defaultLow is zero).  
  ■ overrideHigh: The default value for the cube variable (used if defaultHigh is zero). |
| CalcScripts    | This child element is not used.                                             |
| AnonymousCalcBlock | This child element is not used.                                      |

**Syntax**

```xml
<CubeTable _
  extName=""
  type=""
  isCacheable=""
  cacheExpiry=""
```
8.12 XML Elements: Physical Dimension

The physical dimension element corresponds to the physical dimension object in the Physical layer.

The physical dimension tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>defaultMember</td>
<td>The default member for this physical dimension.</td>
</tr>
</tbody>
</table>

The physical dimension tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchies</td>
<td>References the hierarchies that belong to this physical dimension.</td>
</tr>
<tr>
<td></td>
<td>Each hierarchy is contained in a sub-element called RefPhysicalHierarchy.</td>
</tr>
</tbody>
</table>
### Syntax

```xml
<PhysicalDimension _
    defaultMember=""/>
<Description />
<Hierarchies>
    <RefPhysicalHierarchy ...
        />
</Hierarchies>
<UserAttributes>
    <RefCubeColumn ...
        />
</UserAttributes>
<AlternateHierarchyNames>
    <Name ...
        />
</AlternateHierarchyNames>
</PhysicalDimension>
```

### Example

```xml
<PhysicalDimension name="Months"
    parentName="&quot;03 - Essbase Sample El&quot;&quot;
    parentId="3039:55589" parentUid="80cb9c61-0bf8-0000-714b-e31d00000000"
    id="3064:55587" uid="80cb9c61-0bf8-0000-714b-e31d00000000">
    <Hierarchies>
        <RefPhysicalHierarchy id="3041:55581"
            uid="80cb9c5b-0be1-0000-714b-e31d00000000"
            qualifiedName="&quot;03 - Essbase Sample El&quot;&quot; />
    </Hierarchies>
    <UserAttributes>
        <RefCubeColumn id="3043:55696"
            uid="80cb9ccf-0be3-0000-714b-e31d00000000"
            qualifiedName="&quot;03 - Essbase Sample El&quot;&quot; />
    </UserAttributes>
    <AlternateHierarchyNames>
        <Name />
        <![CDATA[Calendar]]>
        <Name />
        <![CDATA[Fiscal]]>
    </AlternateHierarchyNames>
</PhysicalDimension>
```

### Table 8–20 (Cont.) Physical Dimension Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserAttributes</td>
<td>References the dimension properties for this physical dimension. Each dimension property is contained in a sub-element called RefCubeColumn.</td>
</tr>
<tr>
<td>AlternateHierarchyNames</td>
<td>For Essbase physical dimension objects, lists the names of any alternate hierarchies that belong to this dimension. Each hierarchy name is listed in a sub-element called Name.</td>
</tr>
</tbody>
</table>
8.13 XML Elements: Physical Hierarchy

The physical hierarchy element corresponds to the physical hierarchy object in the Physical layer.

The physical hierarchy tag includes the following attributes:

### Table 8–21 Physical Hierarchy Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memberType</td>
<td>Values are Unknown, Regular, All, Formula, Measure. Default value is Unknown.</td>
</tr>
<tr>
<td>extName</td>
<td>The external name for the physical hierarchy.</td>
</tr>
<tr>
<td>type</td>
<td>The type of physical hierarchy. Options include FULLY BALANCED, UNBALANCED, RAGGED BALANCED, and NETWORK. This option corresponds to the Hierarchy Type option in the Physical Hierarchy dialog in the Administration Tool.</td>
</tr>
<tr>
<td>dimType</td>
<td>The dimension type (for example, Measure Dimension) for the dimension to which this hierarchy belongs.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this physical hierarchy.</td>
</tr>
<tr>
<td>isTimeDim</td>
<td>If set to true, indicates that this physical dimension is a time dimension.</td>
</tr>
<tr>
<td>dimUniqueName</td>
<td>The name of the dimension to which this hierarchy belongs.</td>
</tr>
<tr>
<td>isMemberAliasUnique</td>
<td>Indicates whether member names (including aliases) are unique in a given hierarchy so that the Oracle BI Server can take advantage of specific MDX syntax to optimize performance. This option corresponds to the Use unqualified member name for better performance option in the Physical Hierarchy dialog in the Administration Tool.</td>
</tr>
</tbody>
</table>

The physical hierarchy tag includes the following child elements:

### Table 8–22 Physical Hierarchy Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>For parent-child (value) hierarchies. References the set of cube columns that belong to this physical hierarchy. Each cube column is contained in a sub-element called RefCubeColumn.</td>
</tr>
<tr>
<td>MemberKey</td>
<td>For parent-child (value) hierarchies. References the cube column that is the member key for this hierarchy. The member key column is contained in a sub-element called RefCubeColumn.</td>
</tr>
<tr>
<td>ParentKey</td>
<td>For parent-child (value) hierarchies. References the cube column that is the parent key for this hierarchy. The parent key column is contained in a sub-element called RefCubeColumn.</td>
</tr>
<tr>
<td>Levels</td>
<td>For level-based hierarchies. References the set of physical levels that belong to this physical hierarchy. Each level is contained in a sub-element called RefPhysicalLevel.</td>
</tr>
</tbody>
</table>
Syntax

```xml
<PhysicalHierarchy _
   memberType="" extName=""
   type=""
dimType=""
rowCount=""
isTimeDim=""
dimUniqueName=""
isMemberAliasUnique="">
   <Description />
   <Columns>
      <RefCubeColumn ...
   </Columns>
   <MemberKey>
      <RefCubeColumn ...
   </MemberKey>
   <ParentKey>
      <RefCubeColumn ...
   </ParentKey>
   <Levels>
      <RefPhysicalLevel ___
   </Levels>
</PhysicalHierarchy>
```

Example

```xml
<PhysicalHierarchy name="Employees"
   parentName="04 - Essbase Sample E2 (Default Import)"
   parentId="3039:73265" parentUid="80cbf73c-0bdf-0000-714b-e31d00000000"
   id="3041:73255" uid="80cbf732-0be1-0000-714b-e31d00000000"
   memberType="all"
   extName="Employees"
   type='VALUE' dimType='Other' dimUniqueName="Employees"
isMemberAliasUnique='true'>
   <Columns>
      <RefCubeColumn id="3043:73365"
         uid="80cbf7a0-0be3-0000-714b-e31d00000000"
         qualifiedName="04 - Essbase Sample E2 (Default Import)"
         BISAMPLE=quote;...quote;Sample&quote;"/
      <RefCubeColumn id="3043:73367"
         uid="80cbf7a2-0be3-0000-714b-e31d00000000"
         qualifiedName="04 - Essbase Sample E2 (Default Import)"
         BISAMPLE=quote;...quote;Employee Name&quote;"/
   ...
   </Columns>
   <MemberKey>
      <RefCubeColumn id="3043:73365"
         uid="80cbf7a0-0be3-0000-714b-e31d00000000"
         qualifiedName="04 - Essbase Sample E2 (Default Import)"
         BISAMPLE=quote;...quote;Employee Name&quote;"/
   </MemberKey>
   <ParentKey>
      <RefCubeColumn id="3043:74681"
         uid="80cbfaa4-0be3-0000-714b-e31d00000000"
         qualifiedName="04 - Essbase Sample E2 (Default Import)"
         BISAMPLE=quote;...quote;Employees Parent Key&quote;"/
   </ParentKey>
   <Levels>
```
8.14 XML Elements: Physical Level

The physical level element corresponds to the hierarchical levels for a physical cube table object in the Physical layer.

The physical level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>levelNum</td>
<td>Identifies the order of this physical level in the hierarchy.</td>
</tr>
<tr>
<td>extName</td>
<td>Fully qualified name for the level.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this physical level.</td>
</tr>
</tbody>
</table>

The physical level tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LevelKey</td>
<td>References the key column for this physical level. The key column is contained in a sub-element called RefCubeColumn.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<PhysicalLevel ...
    levelNumber=""
    extName=""
    rowCount=""
><Description />
    <LevelKey>
    <RefCubeColumn ...
</LevelKey>
</PhysicalLevel>
```

Example

```xml
<PhysicalLevel name='Brands'
    parentName='03 - Essbase Sample E1'&quot;.&quot;BISAMPLE
    &quot;.&quot;Sample&quot;.&quot;Products&quot;.&quot;
    parentId='3041:55576' parentUid='80cb9c56-0be1-0000-714b-e31d00000000'
    id='3045:55742' uid='80cb9cfc-0be5-0000-714b-e31d00000000'
    levelNum='1' extName='Brands'>
    <LevelKey>
```
8.15 XML Elements: Cube Column

The cube column element corresponds to the cube column of a physical cube object in the Physical layer.

The cube column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The external name for the cube column. This attribute is required if the same name (such as STATE) is used in multiple hierarchies.</td>
</tr>
<tr>
<td>aggrType</td>
<td>The type of aggregation on the column, if any.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the cube column, such as VARCHAR.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the column. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the cube column.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the column.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The latest number of rows retrieved for this cube column.</td>
</tr>
<tr>
<td>lastUpdated</td>
<td>The time at which row counts were last updated for this cube column.</td>
</tr>
<tr>
<td>dimPropType</td>
<td>For Essbase dimensional columns only. Indicates the property type of the cube column. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ uda: User Defined Attribute (UDA) type.</td>
</tr>
<tr>
<td></td>
<td>■ memnor: Memnor type, used for outline sorts in the logical layer. Imported at the lowest level of each dimension. Called Outline Sort in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>■ attribute: Attribute type, for attribute dimensions.</td>
</tr>
<tr>
<td></td>
<td>■ other: The type is different than those listed, or unknown.</td>
</tr>
<tr>
<td></td>
<td>■ none: The column does not have a property type. Note that this type is mapped to Member Alias in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>■ ancestorRef: References the ancestor of a dimension.</td>
</tr>
<tr>
<td></td>
<td>■ memberKey: The column is a member key.</td>
</tr>
<tr>
<td></td>
<td>■ levelNumber: The column is the lowest member of the hierarchy. Called Leaf in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>■ genNumber: The column is the root member of the hierarchy. Called Root in the Administration Tool.</td>
</tr>
<tr>
<td></td>
<td>■ parentRef: References the parent of a dimension.</td>
</tr>
</tbody>
</table>

The cube column tag includes the following child elements:
### Syntax

```
<CubeColumn _
   extName=""
   aggrType=""
   dataType=""
   nullable=""
   precision=""
   scale=""
   rowCount=""
   lastUpdated=""
   dimPropType="">
   <Description />
   <AttributeHierarchy>
     <RefPhysicalHierarchy/>
   </AttributeHierarchy>
   <PhysicalLevel>
     <RefPhysicalLevel/>
   </PhysicalLevel>
</CubeColumn>
```

### Example

```
<CubeColumn name="Month Names"
   parentName="&quot;03 - Essbase Sample E1&quot;.&quot;BISAMPLE&quot;.
   ..&quot;Sample&quot;"
   parentId="3039:55589" parentUid="80cb9c63-0bdf-0000-714b-e31d00000000"
   id="3043:55660" uid="80cb9caa-0be3-0000-714b-e31d00000000"
   extName="Month Names" dataType="VARCHAR" precision="128"
   nullable="true" dimPropType='none'>
   <PhysicalLevel>
     <RefPhysicalLevel id="3045:55757"
        uid="80cb9d0b-0be5-0000-714b-e31d00000000"
        qualifiedName="&quot;03 - Essbase Sample E1&quot;.&quot;BISAMPLE&quot;.
        ..&quot;Sample&quot;.&quot;Month of Year&quot;.&quot;Month Names&quot;"/>
   </PhysicalLevel>
</CubeColumn>
```

### 8.16 XML Elements: Analytic Workspace

The Analytic Workspace element corresponds to the Oracle OLAP Analytic Workspace (AW) object in the Physical layer.

The Analytic Workspace tag includes the following attributes:
The Analytic Workspace tag includes the following child elements:

**Table 8–27 Analytic Workspace Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of the Analytic Workspace.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<AnalyticWorkspace _
    version="">
    <Description />
    <DynamicName>
        <RefVariable _/>
    </DynamicName>
</AnalyticWorkspace>
```

**Example**

```xml
<AnalyticWorkspace name="GLOBAL"
    parentName="Demo AW";
    parentId="3025:8" parentUid="000000ad-0bd1-0000-714b-e31d00000000"
    id="3049:9" uid="000000ae-0be9-0000-714b-e31d00000000"
    version="10">
</AnalyticWorkspace>
```

### 8.17 XML Elements: AW Cube

The AW cube element corresponds to the Oracle OLAP cube object in the Physical layer.

The AW cube tag includes the following attributes:

**Table 8–29 AW Cube Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the cube in physical SQL queries.</td>
</tr>
<tr>
<td>isCacheable</td>
<td>If set to true, indicates that this table is included in the Oracle BI Server query cache.</td>
</tr>
<tr>
<td>CacheExpiry</td>
<td>How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire.</td>
</tr>
<tr>
<td>isCacheEventTable</td>
<td>If set to true, indicates that this table has been registered as an Oracle BI Server event polling table.</td>
</tr>
<tr>
<td>pollFreq</td>
<td>The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds.</td>
</tr>
</tbody>
</table>
The AW cube tag includes the following child elements:

**Table 8–30 AW Cube Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>References the Oracle OLAP columns that belong to this Oracle OLAP cube. Each Oracle OLAP column is contained in a sub-element called RefAWColumn.</td>
</tr>
<tr>
<td>AggrConnPool</td>
<td>If this Oracle OLAP cube is an aggregate table, this element references the connection pool used for the aggregate. The connection pool is contained in a sub-element called RefConnectionPool.</td>
</tr>
<tr>
<td>CompositeInfo</td>
<td>References dimensions for the cube. Contains the sub-element called Composite, which has one attribute called name and a sub-element called Dimension. Each dimension is contained in the Dimension sub-element.</td>
</tr>
</tbody>
</table>

**Syntax**

```
<AWCube ...
```
Example

<AWCube name="Units Cube"
    parentName="Demo AWquot;GLOBALquot;GLOBALquot;" 
    parentId="3049:9" parentUid="000000ae-0be9-0000-714b-e31d00000000" 
    id="3054:10" uid="000000af-0bee-0000-714b-e31d00000000" 
    extName="Units Cube">
    <Columns>
        <RefAWColumn id="3060:11" 
            uid="000000b5-0bf4-0000-714b-e31d00000000" 
            qualifiedName="Demo AWquot;GLOBALquot;GLOBALquot;Units Cubequot;Unitsquot;"/>
        <RefAWColumn id="3060:12" 
            uid="000000b6-0bf4-0000-714b-e31d00000000" 
            qualifiedName="Demo AWquot;GLOBALquot;GLOBALquot;Units Cubequot;Salesquot;"/>
    </Columns>
    <CompositeInfo>
        <Composite name="UNITS_CUBE_COMPOSITE">
            <Dimension><![CDATA[CUSTOMER]]></Dimension>
            <Dimension><![CDATA[PRODUCT]]></Dimension>
            <Dimension><![CDATA[CHANNEL]]></Dimension>
        </Composite>
    </CompositeInfo>
</AWCube>
8.18 XML Elements: AW Dimension

The AW dimension element corresponds to the Oracle OLAP dimension object in the Physical layer.

The AW dimension tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The dimension type.</td>
</tr>
<tr>
<td>isCacheable</td>
<td>If set to true, indicates that this dimension is included in the Oracle BI Server query cache.</td>
</tr>
<tr>
<td>CacheExpiry</td>
<td>If set to true, indicates that this dimension is included in the Oracle BI Server query cache.</td>
</tr>
<tr>
<td>isCacheEventTable</td>
<td>How long table entries should persist in the query cache. A value of 0 indicates that cache entries do not expire.</td>
</tr>
<tr>
<td>isOverrideCacheProp</td>
<td>For alias tables only. If set to true, indicates that the alias table has its own cache properties that override the cache properties of the source table.</td>
</tr>
<tr>
<td>pollFreq</td>
<td>If set to true, indicates that this table has been registered as an Oracle BI Server event polling table.</td>
</tr>
<tr>
<td>x</td>
<td>The polling frequency, in seconds. Only applies if this table is an Oracle BI Server event polling table. The default value is 3600 seconds.</td>
</tr>
<tr>
<td>y</td>
<td>The x coordinate of the table when it is shown in the physical diagram.</td>
</tr>
<tr>
<td>rowCount</td>
<td>The y coordinate of the table when it is shown in the physical diagram.</td>
</tr>
<tr>
<td>lastUpdated</td>
<td>The time at which row counts were last updated for this Oracle OLAP dimension.</td>
</tr>
<tr>
<td>hints</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>isSysGen</td>
<td>If set to true, indicates that an internal utility has generated this Oracle OLAP dimension.</td>
</tr>
<tr>
<td>isDDLAggr</td>
<td>Indicates whether this Oracle OLAP dimension is an aggregate table.</td>
</tr>
<tr>
<td>isTimeDim</td>
<td>If set to true, indicates that this Oracle OLAP dimension is a time dimension.</td>
</tr>
<tr>
<td>isRagged</td>
<td>If set to true, indicates that this dimension hierarchy is unbalanced. An unbalanced hierarchy is one that contains levels that are not at the same depth.</td>
</tr>
<tr>
<td>isSkipped</td>
<td>If set to true, indicates that this dimension hierarchy is a skip-level hierarchy. A skip-level hierarchy is one in which there are members that do not have a value for a particular parent level.</td>
</tr>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the dimension in physical SQL queries.</td>
</tr>
</tbody>
</table>

The AW dimension tag includes the following child elements:
**Table 8–32  AW Dimension Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AggrConnPool</td>
<td>If this AW dimension is an aggregate table, this element references the connection pool used for the aggregate. The connection pool is contained in a sub-element called RefConnectionPool.</td>
</tr>
<tr>
<td>AWObjects</td>
<td>References the Oracle OLAP objects, such as a list of hierarchies, associated with this dimension. Each object is contained in a sub-element called AWObject. This sub-element includes the following attributes:</td>
</tr>
<tr>
<td></td>
<td>■ name</td>
</tr>
<tr>
<td></td>
<td>■ type</td>
</tr>
<tr>
<td></td>
<td>■ dataType</td>
</tr>
<tr>
<td></td>
<td>■ nullable</td>
</tr>
<tr>
<td></td>
<td>■ precision</td>
</tr>
<tr>
<td></td>
<td>■ scale</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<AWDimension _
  type=""
  isCacheable=""
  cacheExpiry=""
  isCacheEventTable=""
  isOverrideCacheProp=""
  pollFreq=""
  x=""
  y=""
  rowCount=""
  lastUpdated=""
  isSysGen=""
  isDDLAggr=""
  isTimeDim=""
  extName="">
  <Description />
  <AggrConnPool>
    <RefConnectionPool …/>  
  </AggrConnPool>
  </AWObjects>
  <AWObject name="" type="" dataType="" ...
  <AWObject name="" type="" dataType="" ...
</AWDimension>
```

**Example**

```xml
<AWDimension name="Customer"
  parentName="Demo AW"&quot;.GLOBAL&quot;.GLOBAL"
  parentId="3049:9" parentUid="000000ae-0be9-0000-714b-e31d00000000"
  id="3052:91" uid="000000b3-0bec-0000-714b-e31d00000000"
  type='none' extName='CUSTOMER'>
  <AWObjects>
    <AWObject type="" HIER_LIST" name="CUSTOMER_HIERLIST" dataType="VARCHAR" precision='100'/>
    <AWObject type="" LEVEL_LIST" name="CUSTOMER_LEVELLIST" dataType="VARCHAR" precision='100'/>
</AWDimension>
```
8.19 XML Elements: AW Hierarchy

The AW hierarchy element corresponds to the Oracle OLAP hierarchy object in the Physical layer.

The AW hierarchy tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of Oracle OLAP hierarchy (value or level).</td>
</tr>
<tr>
<td>isDefault</td>
<td>If set to true, indicates that this is the default hierarchy for the Oracle OLAP dimension.</td>
</tr>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the hierarchy in physical SQL queries.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the Oracle OLAP hierarchy, such as VARCHAR.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the Oracle OLAP hierarchy. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the Oracle OLAP hierarchy.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the Oracle OLAP hierarchy.</td>
</tr>
</tbody>
</table>

The AW hierarchy tag includes the following child elements:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levels</td>
<td>For level-based hierarchies only. References the level or levels for this Oracle OLAP hierarchy. Each level is contained in a sub-element called RefAWLevel.</td>
</tr>
<tr>
<td>Columns</td>
<td>For value-based hierarchies only. References the column or columns for this Oracle OLAP hierarchy. Each column is contained in a sub-element called RefAWColumn.</td>
</tr>
<tr>
<td>MemberKey</td>
<td>References the member key (for parent-child hierarchies). The member key is contained in a sub-element called RefAWColumn.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<AWHierarchy ...
    type="*
    isDefault="*
```
8.20 XML Elements: AW Level

The AW level element corresponds to the Oracle OLAP level object in the Physical layer.

The AW level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the level in physical SQL queries.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the Oracle OLAP level, such as VARCHAR.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the Oracle OLAP level. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the Oracle OLAP level.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the Oracle OLAP level.</td>
</tr>
</tbody>
</table>

The AW level tag includes the following child elements:
### Syntax

```
<AWLevel ... extName="" dataType="" nullable="" precision="" scale="">
   <Description />
   <Columns>
      <RefAWColumn .../>
   </Columns>
   <LevelKey>
      <RefAWColumn .../>
   </LevelKey>
</AWLevel>
```

### Example

```
<AWLevel name="All Years"
   parentName=""parentName":"Demo AW"&quot;.&quot;GLOBAL&quot;.&quot;GLOBAL&quot;.
   &quot;Time&quot;.&quot;Calendar Year&quot;"
   parentId="3056:136" parentUid="0000012d-0bf0-0000-714b-e31d00000000"
   id="3058:142" uid="00000133-0bf2-0000-714b-e31d00000000"
   extName="ALL_YEARS" dataType="UNKNOWN">
   <Columns>
      <RefAWColumn id="3060:49"
         uid="00000009-0bf4-0000-714b-e31d00000000"
         qualifiedName=""qualifiedName":"Demo AW"&quot;.&quot;GLOBAL&quot;.&quot;GLOBAL&quot;.
         &quot;Time&quot;.&quot;CALENDAR_YEAR - All Years - Long Description"
         &quot;"
         />
      <RefAWColumn id="3060:50"
         uid="000000da-0bf4-0000-714b-e31d00000000"
         qualifiedName=""qualifiedName":"Demo AW"&quot;.&quot;GLOBAL&quot;.&quot;GLOBAL&quot;.
         &quot;Time&quot;.&quot;Calendar Year - All Years&quot;
         &quot;"/>
   </Columns>
   <LevelKey>
      <RefAWColumn id="3060:50"
         uid="000000da-0bf4-0000-714b-e31d00000000"
         qualifiedName=""qualifiedName":"Demo AW"&quot;.&quot;GLOBAL&quot;.&quot;GLOBAL&quot;.
         &quot;Time&quot;.&quot;Calendar Year - All Years&quot;"/>
   </LevelKey>
</AWLevel>
```

### 8.21 XML Elements: AW Column

The AW column element corresponds to the AW column object in the Physical layer. The AW column tag includes the following attributes:

#### Table 8–36 AW Level Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>References the column or columns for this Oracle OLAP level. Each column is contained in a sub-element called RefAWColumn.</td>
</tr>
<tr>
<td>LevelKey</td>
<td>References the key column for this Oracle OLAP level. The key column is contained in a sub-element called RefAWColumn.</td>
</tr>
</tbody>
</table>
The AW column tag does not include any child elements.

**Syntax**

```xml
<AWColumn ... extName="" isParentKey="" dataType="" nullable="" precision="" scale="">
<Description />
</AWColumn>
```

**Example**

```xml
<AWColumn name="Sales"
    parentName="Demo AW"&quot;GLOBAL&quot;.&quot;GLOBAL&quot;.&quot;Units Cube"
    parentId="3054:10" parentUid="000000af-0bee-0000-714b-e31d00000000"
    id="3060:12" uid="000000b6-0bf4-0000-714b-e31d00000000"
    extName="UNITS_CUBE_SALES" dataType="DOUBLE" nullable="true">
```

### Table 8–37 AW Column Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extName</td>
<td>The physical name that is used when referencing the column in physical SQL queries.</td>
</tr>
<tr>
<td>isParentKey</td>
<td>Indicates whether this Oracle OLAP column is a parent key for a level.</td>
</tr>
<tr>
<td>dataType</td>
<td>The data type of the Oracle OLAP column, such as VARCHAR.</td>
</tr>
<tr>
<td>scale</td>
<td>The scale of the data type of the Oracle OLAP column.</td>
</tr>
<tr>
<td>nullable</td>
<td>If set to true, indicates that null values are allowed for the Oracle OLAP column. This allows null values to be returned to the user, which is expected with certain functions and with outer joins.</td>
</tr>
<tr>
<td>precision</td>
<td>The precision of the data type of the Oracle OLAP column.</td>
</tr>
</tbody>
</table>
This chapter provides Oracle BI Server XML API reference information for presentation elements. Presentation elements typically correspond to objects in the Presentation layer of the Oracle BI repository, or in other words, the objects that appear in the left-hand pane of the Oracle BI Administration Tool.

This chapter contains the following topics:

- Section 9.1, "XML Elements: Presentation Catalog"
- Section 9.2, "XML Elements: Presentation Schema"
- Section 9.3, "XML Elements: Presentation Table"
- Section 9.4, "XML Elements: Presentation Column"
- Section 9.5, "XML Elements: Presentation Hierarchy"
- Section 9.6, "XML Elements: Presentation Level"

### 9.1 XML Elements: Presentation Catalog

The presentation catalog element corresponds to the subject area object in the Presentation layer.

The presentation catalog tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isExportKeys</td>
<td>If set to true, indicates that logical keys for this subject area are exposed to other applications.</td>
</tr>
<tr>
<td>hasDispName</td>
<td>Indicates whether this subject area has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this subject area.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this subject area has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this subject area.</td>
</tr>
<tr>
<td>isAutoAggr</td>
<td>If set to true, indicates that auto aggregation is enabled. Auto aggregation is the process by which the Oracle BI Server infers the GROUP BY from the attributes projected in the query. Typically, auto aggregation is always enabled for queries against the Presentation layer.</td>
</tr>
</tbody>
</table>

The presentation catalog tag includes the following child elements:
### Table 9–2 Presentation Catalog Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefBusinessModel</td>
<td>References the business model for this subject area.</td>
</tr>
<tr>
<td>DefaultFactCol</td>
<td>References the implicit fact column for this subject area, if one has been set. This column is used to specify a default join path between dimension tables when there are several possible alternatives or contexts. The implicit fact column is contained in a sub-element called RefLogicalColumn.</td>
</tr>
<tr>
<td>Aliases</td>
<td>Lists any aliases that have been defined for this subject area. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias is contained in a sub-element called Alias, with one attribute called name.</td>
</tr>
<tr>
<td>Tables</td>
<td>References the presentation tables that belong to this subject area. Each presentation table is contained in a sub-element called RefPresentationTable.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in a sub-element called Expr.</td>
</tr>
</tbody>
</table>

### Syntax

```xml
<PresentationCatalog _
    isExportKeys=""
    isAutoAggr=""
    hasDispName=""
    dispName=""
    hasDispDescription=""
    dispDescription"

<Description />
<RefBusinessModel .../>
<DefaultFactCol>
    <RefLogicalColumn .../>
</DefaultFactCol>
</DefaultFactCol>
<Aliases ...

<Tables>
    <RefPresentationTable ...
</Tables>
<VisibilityFilter>
    <Expr ...
</VisibilityFilter>
</PresentationCatalog>
```

### Example

```xml
<PresentationCatalog name="C - Sample Headcount"
    id="4004:20154" uid="80cad5f1-0fa4-0000-714b-e31d00000000"
    hasDispName="false" hasDispDescription="false">
    <Description>
        <![CDATA[Sample Subject Area for Headcount analysis. ]]>
    </Description>
    <RefBusinessModel id="2000:11614"
        uid="80cab573-07d0-0000-714b-e31d00000000"
```
9.2 XML Elements: Presentation Schema

The presentation schema element is not used.

9.3 XML Elements: Presentation Table

The presentation table element corresponds to the presentation table object in the Presentation layer.

The presentation table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation table has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this presentation table.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this presentation table has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this presentation table.</td>
</tr>
</tbody>
</table>

The presentation table tag includes the following child elements:
### Table 9–4 Presentation Table Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>Lists any aliases that have been defined for this presentation table. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias is contained in a sub-element called Alias, with one attribute called name.</td>
</tr>
<tr>
<td>Columns</td>
<td>References the presentation columns that belong to this presentation table. Each presentation column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in a sub-element called Expr.</td>
</tr>
<tr>
<td>ContentFolders</td>
<td>References any child presentation tables that have been designated for this presentation table. Child presentation tables appear as nested folders in Answers and BI Composer. Note that the folders only appear to be nested - they are not actually nested in terms of drill-down, and the qualified names of the objects remain the same. Each child presentation table is contained in a sub-element called RefPresentationTable.</td>
</tr>
</tbody>
</table>

**Syntax**

```
<PresentationTable ...
    hasDispName=""
    dispName=""
    hasDispDescription=""
    dispDescription=""

    <Description />

    <Aliases>
      <Alias name=""/>
      <Alias name=""/>
    </Aliases>

    <Columns>
      <RefPresentationColumn .../>
    </Columns>

    <VisibilityFilter>
      <Expr .../>
    </VisibilityFilter>

    <ContentFolders>
      <RefPresentationTable .../>
    </ContentFolders>

</PresentationTable>
```

**Example**

```
<PresentationTable name="Offices"
    parentName="A - Sample Sales"
    parentUid="80cab7ef-0fa4-0000-714b-e31d00000000"
    id="4008:12457" uid="80cab839-0fa8-0000-714b-e31d00000000"
    dispName="VALUEOF(NQ_SESSION.CN_A_-_Sample_Sales_Offices)"
    hasDispName="true" hasDispDescription="false">

    <Description />

</PresentationTable>
```
9.4 XML Elements: Presentation Column

The presentation column element corresponds to the presentation column object in the Presentation layer.

The presentation column tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>overrideLogicalName</td>
<td>Indicates whether the presentation column has been configured to always use the logical column name. This option corresponds to Use Logical Column Name in the Administration Tool. A false value for this attribute indicates that this option has been selected.</td>
</tr>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation column has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this presentation column.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this presentation column has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this presentation column.</td>
</tr>
</tbody>
</table>

The presentation column tag includes the following child elements:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefLogicalColumn</td>
<td>References the logical column upon which this presentation column is based.</td>
</tr>
</tbody>
</table>
### Syntax

```xml
<PresentationColumn
    overrideLogicalName=""
    hasDispName=""
    dispName=""
    hasDispDescription=""
    dispDescription="">
    <Description />
    <RefLogicalColumn ...
    <Aliases>
        <Alias name="" />
        <Alias name="" />
    </Aliases>
    <VisibilityFilter>
        <Expr ...
    </VisibilityFilter>
</PresentationColumn>
```

### Example

```xml
<PresentationColumn name="T01 Per Name Week (Rel Only)"
    parentName="F - Sample Essb Federated"..&quot;Time""
    parentId="4008:54644" parentUid="80cb981c-0fa8-0000-714b-e31d00000000"
    id="4010:54646" uid="80cb981e-0faa-0000-714b-e31d00000000"
    hasDispName="false" hasDispDescription="false" overrideLogicalName="false">
    <Description>
        <![CDATA[Help : Period Name Week. Stores the week information in the format YYYY Week Number. Eg :- 2006 Week 40]]>
    </Description>
    <RefLogicalColumn id="2006:53537"
        uid="80cb94b1-07d6-0000-714b-e31d00000000"
        qualifiedName="&quot;3 - Sample Federated&quot;..&quot;D0 Time&quot;..&quot;T01 Per Name Week (Rel Only)&quot;">
    <Aliases>
        <Alias name="T01 Per Name Week" />
    </Aliases>
</PresentationColumn>
```

### 9.5 XML Elements: Presentation Hierarchy

The presentation hierarchy element corresponds to the presentation hierarchy object in the Presentation layer.

The presentation hierarchy tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>Lists any aliases that have been defined for this presentation column. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias is contained in a sub-element called Alias, with one attribute called name.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in a sub-element called Expr.</td>
</tr>
</tbody>
</table>

### Table 9–6 (Cont.) Presentation Column Child Elements

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>Lists any aliases that have been defined for this presentation column. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias is contained in a sub-element called Alias, with one attribute called name.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in a sub-element called Expr.</td>
</tr>
</tbody>
</table>
The presentation hierarchy tag includes the following child elements:

### Table 9–8  Presentation Hierarchy Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation hierarchy has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this presentation hierarchy.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this presentation hierarchy has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this presentation hierarchy.</td>
</tr>
</tbody>
</table>

### Table 9–8  Presentation Hierarchy Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levels</td>
<td>References the presentation levels for this presentation hierarchy. Each presentation level is contained in a sub-element called RefPresentationLevel.</td>
</tr>
<tr>
<td>Alias</td>
<td>Lists any aliases that have been defined for this presentation hierarchy. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias is contained in a sub-element called Alias, with one attribute called name.</td>
</tr>
<tr>
<td>DisplayColumns</td>
<td>For parent-child hierarchies only. This element references the columns designated to be used for display for this parent-child hierarchy. Each display column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>Dimension</td>
<td>References the logical dimension upon which this presentation hierarchy is based. The dimension is contained in a sub-element called RefDimension.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>Contains the expression specified to control the visibility of this object in Answers and BI Composer. This element corresponds to the Hide object if field in the Administration Tool. The expression is contained in a sub-element called Expr.</td>
</tr>
</tbody>
</table>

### Syntax

```
<PresentationHierarchy
  hasDispName=""
  displayName=""
  hasDispDescription=""
  dispDescription="">
  <Description />
  <Levels>
    <RefPresentationLevel …/>
  </Levels>
  <Aliases>
    <Alias name=""/>
    <Alias name=""/>
  </Aliases>
  <DisplayColumns>
    <RefPresentationColumn …/>
```

9.6 XML Elements: Presentation Level

The presentation level element corresponds to the presentation level object in the Presentation layer.

The presentation level tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this presentation level has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this presentation level.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this presentation level has a custom description.</td>
</tr>
<tr>
<td>dispDescription</td>
<td>The value of the custom description, if one exists for this presentation level.</td>
</tr>
</tbody>
</table>

The presentation level tag includes the following child elements:
### Table 9–10  Presentation Level Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefLogicalLevel</td>
<td>References the logical level upon which this presentation level is based.</td>
</tr>
<tr>
<td>DrillTo</td>
<td>References the levels for cross-dimension drill-down operations, as in the case of a Region dimension with a City level that can have a &quot;drill to&quot; level Year in the Time dimension.</td>
</tr>
<tr>
<td>DisplayColumns</td>
<td>References the columns that have been designated to be used for display for this presentation level (on drill-down). Each column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>Aliases</td>
<td>Lists any aliases that have been defined for this presentation level. Aliases are created when presentation objects are renamed to ensure that any references to the old names still work in logical SQL queries. Each alias is contained in a sub-element called Alias, with one attribute called name.</td>
</tr>
<tr>
<td>VisibilityFilter</td>
<td>This child element is not used.</td>
</tr>
</tbody>
</table>

### Syntax

```xml
<PresentationLevel _
  hasDisplayName="" displayName="" hasDispDescription="" dispDescriptions="">
  <Description />
  <RefLogicalLevel _/>
  <DrillTo>
    <RefPresentationLevel _/>
    <RefPresentationLevel _/>
  </DrillTo>
  <DisplayColumns>
    <RefPresentationColumn _/>
    <RefPresentationColumn _/>
  </DisplayColumns>
  <Aliases>
    <Alias name="/">
    <Alias name="/">
  </Aliases>
</PresentationLevel>
```

### Example

```xml
<PresentationLevel name="Ship to Region"
  parentName="&quot;A - Sample Sales&quot;.&quot;Orders&quot;.&quot;Ship To Region Hier&quot;"
  parentIId="4028:32353" parentUId="80cb38ef-0fbc-0000-714b-e31d00000000"
  id="4030:32356" uid="80cb38f3-0fbe-0000-714b-e31d00000000"
  dispName="VALUEOF(NQ_SESSION.CN_Orders_Ship_To_Region_Hier_Ship_to_Region)"
  hasDispName="true" hasDispDescription="false">
  <RefLogicalLevel id="2025:32260"
    uid="80cb38a5-07e9-0000-714b-e31d00000000"
    qualifiedName="&quot;1 - Sample App&quot;.&quot;H9 Orders&quot;.&quot;Ship To Region&quot;"/>
  <DisplayColumns>
    <RefPresentationColumn id="4010:32350"/>
  </DisplayColumns>
</PresentationLevel>
```
uid="80cb38ec-0faa-0000-714b-e31d00000000"
qualifiedName='"A - Sample Sales".."Ship To Regions" ."R50 Region"';
</DisplayColumns>
</Aliases>
</PresentationLevel>
This chapter provides Oracle BI Server XML API reference information for Oracle Marketing Segmentation elements, security elements, and miscellaneous Oracle BI repository elements that do not fall in other categories.

This chapter contains the following topics:

- Section 10.1, "XML Marketing Elements"
- Section 10.2, "XML Security Elements"
- Section 10.3, "Other XML Elements"

## 10.1 XML Marketing Elements

Marketing elements are typically those elements that are related to features for the Oracle Marketing Segmentation product. For information about features and options related to Oracle Marketing Segmentation, see the administrator documentation for the Oracle Marketing Segmentation application.

Marketing elements include:

- Section 10.1.1, "XML Elements: Catalog Link"
- Section 10.1.2, "XML Elements: Target Level"
- Section 10.1.3, "XML Elements: List Catalog"
- Section 10.1.4, "XML Elements: Qualified List Item"
- Section 10.1.5, "XML Elements: Qualifying Key"
- Section 10.1.6, "XML Elements: Sampling Table"
- Section 10.1.7, "XML Elements: Seg Catalog"

### 10.1.1 XML Elements: Catalog Link

The catalog link element corresponds to the conforming dimension link object in the Marketing metadata. A conforming dimension links a fact that contains target-level IDs to a fact that does not contain target-level IDs by navigating along a dimension that is shared by both fact tables.

The catalog link tag includes the following child elements:
The catalog link tag does not include any attributes.

**Syntax**

```xml
<CatalogLink ...>
  <Description />  
  <From>
    <RefPresentationColumn .../>
  </From>
  <To>
    <RefPresentationColumn .../>
  </To>
</CatalogLink>
```

**Example**

```xml
<CatalogLink name="Customer Profile-&gt;Campaign Contacts"
  id="4012:280960" uid="8032f5a3-0fac-0000-714b-e31d00000000">
  <Description>  
    <![CDATA[Used by both Contact and Account Target Levels to filter with "Offered Product_segmentation" catalog.]]>
  </Description>
  <To>
    <RefPresentationColumn id="4010:223709" uid="80312fc3-0faa-0000-714b-e31d00000000"
      qualifiedName="&quot;Campaign Contacts_segmentation&quot;..&quot;Contact Profile&quot;..&quot;ROW_ID&quot;"/>
  </To>
</CatalogLink>
```

### 10.1.2 XML Elements: Target Level

The target level element corresponds to the target level object in the Marketing metadata. A target level is the entity that a marketer wants to count, typically customer types like individuals, businesses, or households.

The target level tag includes the following attributes:

**Table 10–2 Target Level Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDispName</td>
<td>Indicates whether this target level has a custom display name.</td>
</tr>
<tr>
<td>dispName</td>
<td>The value of the custom display name, if one exists for this target level.</td>
</tr>
<tr>
<td>hasDispDescription</td>
<td>Indicates whether this target level has a custom description.</td>
</tr>
</tbody>
</table>
The target level tag includes the following child elements:

**Table 10–3 Target Level Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrimaryQLI</td>
<td>References the primary qualified list item for this target level.</td>
</tr>
<tr>
<td></td>
<td>The qualified list item is contained in a sub-element called RefQualifiedListItem.</td>
</tr>
<tr>
<td>PrimaryPresCatalog</td>
<td>References the primary segmentation catalog for this target level.</td>
</tr>
<tr>
<td></td>
<td>The segmentation catalog is contained in a sub-element called RefPresentationCatalog.</td>
</tr>
<tr>
<td>SamplingTables</td>
<td>References the physical tables used as sampling tables for this target level.</td>
</tr>
<tr>
<td></td>
<td>Each physical table is contained in a sub-element called RefSamplingTable.</td>
</tr>
<tr>
<td>Catalogs</td>
<td>References the set of segmentation catalogs for this target level.</td>
</tr>
<tr>
<td></td>
<td>Each segmentation catalog is contained in a sub-element called RefSegCatalog.</td>
</tr>
<tr>
<td>PresentationCatalog</td>
<td>References the subject area for saved result sets (the saved results catalog).</td>
</tr>
<tr>
<td></td>
<td>The saved results catalog is contained in a sub-element called RefPresentationCatalog.</td>
</tr>
<tr>
<td>TargetColumn</td>
<td>References the Target Id presentation column.</td>
</tr>
<tr>
<td></td>
<td>The Target Id column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>NodeColumn</td>
<td>References the GUID presentation column.</td>
</tr>
<tr>
<td></td>
<td>The GUID column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>PhysicalTable</td>
<td>References the physical table that stores the result sets for this target level.</td>
</tr>
<tr>
<td></td>
<td>The physical table is contained in a sub-element called RefPhysicalTable.</td>
</tr>
<tr>
<td>ConnectionPool</td>
<td>References the connection pool in which this target level is mapped.</td>
</tr>
<tr>
<td></td>
<td>The connection pool is contained in a sub-element called RefConnectionPool.</td>
</tr>
<tr>
<td>HeaderSQL</td>
<td>Contains the physical SQL to insert the Saved Result Header. Used when users save a result set.</td>
</tr>
<tr>
<td>DelHeaderSQL</td>
<td>Contains the physical SQL to delete the saved result set header. Used when users purge saved result sets.</td>
</tr>
<tr>
<td>DelDataSQL</td>
<td>Contains the physical SQL to delete the saved result data set. Used when users purge saved result sets.</td>
</tr>
</tbody>
</table>
Syntax

```xml
<TargetLevel _
  hasDispName=""
  dispName=""
  hasDispDescription=""
  dispDescription="">
  <Description />
  <PrimaryQLI>
    <RefQualifiedListItem _/>
  </PrimaryQLI>
  <PrimaryPresCatalog>
    <RefPresentationCatalog _/>
  </PrimaryPresCatalog>
  <SamplingTables>
    <RefSamplingTable _/>
  </SamplingTables>
  <Catalogs>
    <RefSegCatalog _/>
    <RefSegCatalog _/>
  </Catalogs>
  <PresentationCatalog>
    <RefPresentationCatalog _/>
  </PresentationCatalog>
  <TargetColumn>
    <RefPresentationColumn _/>
  </TargetColumn>
  <NodeColumn>
    <RefPresentationColumn _/>
  </NodeColumn>
  <PhysicalTable>
    <RefPhysicalTable _/>
  </PhysicalTable>
  <ConnectionPool>
    <RefConnectionPool _/>
  </ConnectionPool>
  <HeaderSQL _/>
  <DelHeaderSQL _/>
  <DelDataSQL _/>
</TargetLevel>
```

Example

```xml
<TargetLevel name="Claims"
  id="4014:280967" uid="803421c3-0fae-0000-714b-e31d00000000"
  hasDispName="false" hasDispDescription="false">
  <PrimaryQLI>
    <RefQualifiedListItem id="4018:280971"
      uid="80341f19-0fb2-0000-714b-e31d00000000"
      qualifiedName="&quot;Claims&quot;"/>
  </PrimaryQLI>
  <PrimaryPresCatalog>
    <RefPresentationCatalog id="4004:275727"
      uid="803405cb-0fa4-0000-714b-e31d00000000"
      qualifiedName="&quot;Insurance Claims_Segmentation&quot;"/>
  </PrimaryPresCatalog>
  <SamplingTables>
    <RefSamplingTable id="4024:281052"
      uid="80342e79-0fb8-0000-714b-e31d00000000"
      qualifiedName="&quot;Claims&quot;.&quot;.&quot;M_10C_INSCLM_D&quot;"/>
  </SamplingTables>
```
10.1.3 XML Elements: List Catalog

The list catalog element corresponds to the list catalog object in the Marketing metadata. A list catalog is an object that is used to generate a list of data at the target level.

The list catalog tag includes the following child elements:
The list catalog tag does not include any attributes.

Syntax

```xml
<ListCatalog ...
    <Description />
    <RefPresentationCatalog ...
    <QualifyingKeys>
        <RefQualifyingKey ...
        <RefQualifyingKey ...
    </QualifyingKeys>
</ListCatalog>
```

Example

```xml
<ListCatalog name="Campaign Load - Contacts"
    id="4016:281116" uid="8032fe2c-0fb0-0000-714b-e31d00000000">
    <RefPresentationCatalog id="4004:275725"
        uid="8032fb8c-0fa4-0000-714b-e31d00000000"
        qualifiedName="Campaign Load - Contacts"/>
    <QualifyingKeys>
        <RefQualifyingKey id="4020:281006"
            uid="8032fe3d-0fb4-0000-714b-e31d00000000"
            qualifiedName="Contacts.Qualifying Key#29"/>
        <RefQualifyingKey id="4020:281037"
            uid="8032fe3e-0fb4-0000-714b-e31d00000000"
            qualifiedName="Accounts.Qualifying Key#30"/>
    </QualifyingKeys>
</ListCatalog>
```

10.1.4 XML Elements: Qualified List Item

The qualified list item element corresponds to the qualified list item object in the Marketing metadata. A qualified list item is an entity that is evaluated against segment criteria so that the information related to that entity can be exported in a list file.

The qualified list item tag includes the following attributes:

Table 10–5 Qualified List Item Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacheTable</td>
<td>Indicates whether this qualified list item is cached.</td>
</tr>
</tbody>
</table>

The qualified list item tag includes the following child elements:

The list catalog tag does not include any attributes.
Table 10-6 Qualified List Item Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheCatalog</td>
<td>References the subject area for the cache for this qualified list item. The subject area is contained in a sub-element called RefPresentationCatalog.</td>
</tr>
<tr>
<td>CacheConnPool</td>
<td>References the connection pool for the cached table. The connection pool is contained in a sub-element called RefConnectionPool.</td>
</tr>
<tr>
<td>TargetColumn</td>
<td>References the Target Id presentation column. The Target Id column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>NodeColumn</td>
<td>References the GUID presentation column. The GUID column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
<tr>
<td>PhysicalTable</td>
<td>References the physical table that stores the cached qualified list item. The physical table is contained in a sub-element called RefPhysicalTable.</td>
</tr>
<tr>
<td>DeleteSQL</td>
<td>Contains the physical SQL used to delete the cached results.</td>
</tr>
</tbody>
</table>

Syntax

```
<QualifiedListItem _
   cacheTable="">
   <Description />
   <CacheCatalog>
     <RefPresentationCatalog ...
   </CacheCatalog>
   <CacheConnPool>
     <RefConnectionPool ...
   </CacheConnPool>
   <TargetColumn>
     <RefPresentationColumn ...
   </TargetColumn>
   <NodeColumn>
     <RefPresentationColumn ...
   </NodeColumn>
   <PhysicalTable>
     <RefPhysicalTable ...
   </PhysicalTable>
   <DeleteSQL>
     <![CDATA[DELETE FROM M_C_GENERIC WHERE GUID = '@{guid}']]>
   </DeleteSQL>
</QualifiedListItem>
```

Example

```
<QualifiedListItem name="Service Requests"
   id='4018:6474' uid='80caald3-0fb2-0000-714b-e31d00000000'>
   <DeleteSQL>
     <![CDATA[DELETE FROM M_C_GENERIC WHERE GUID = '@{guid}']]>
   </DeleteSQL>
</QualifiedListItem>
```
10.1.5 XML Elements: Qualifying Key

The qualifying key element corresponds to the qualifying keys that are used in list catalog or segmentation catalog objects in the Marketing metadata. Qualifying keys map qualified list items to presentation columns.

The qualifying key tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefSegCatalog</td>
<td>References the segmentation catalog to which this qualifying key belongs.</td>
</tr>
<tr>
<td>RefListCatalog</td>
<td>References the list catalog to which this qualifying key belongs.</td>
</tr>
<tr>
<td>RefPresentationColumn</td>
<td>References the presentation column for this qualifying key.</td>
</tr>
</tbody>
</table>

The qualifying key tag does not include any attributes.

Syntax

```xml
<QualifyingKey ...
  <Description />
  <RefSegCatalog ../>
  <RefListCatalog ../>
  <RefPresentationColumn ../>
</QualifyingKey>
```

Example

```xml
<QualifyingKey name="MKTG_QK_4020:125971262767661"
  parentName="&quot;Customers&quot;"
  parentId="4018:6474" parentUid="80caa1d3-0fba-0000-714b-e31d00000000"
  id="4020:12597" uid="80cab8c6-0faa-0000-714b-e31d00000000">
  <RefSegCatalog id="4026:12596"
    uid="80cab8c7-0fab-0000-714b-e31d00000000"
    qualifiedName="&quot;Customers&quot;.&quot;Customers_A - Sample Sales&quot;"/>
  <RefPresentationColumn id="4010:12527"
    uid="80cab87f-0faa-0000-714b-e31d00000000"
    qualifiedName="&quot;A - Sample Sales&quot;.&quot;More Customer Objects&quot;..&quot;&quot;
    .&quot;X20 Cust Key (Segmentation)&quot;"/>
</QualifyingKey>
```

10.1.6 XML Elements: Sampling Table

The sampling table element corresponds to the sampling table object in the Marketing metadata. Sampling tables are created and managed as part of target level objects.

The sampling table tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>factor</td>
<td>The sampling factor (for example, 10%).</td>
</tr>
</tbody>
</table>

The sampling table tag includes the following child elements:
Table 10–9 Sampling Table Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefPhysicalTable</td>
<td>References the physical table being used as the sampling table.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<SamplingTable _
   factor="">
   <Description />
   <RefPhysicalTable _/>
</SamplingTable>
```

Example

```xml
<SamplingTable name="M_10C_PERSON_D"
   parentName="Contacts"
   parentId="4014:280966" parentUid="802ba17b-0fae-0000-714b-e31d00000000"
   id="4024:281051" uid="80331464-0fb8-0000-714b-e31d00000000"
   factor="0.1">
   <RefPhysicalTable id="3001:86873"
   uid="002261ab-0bb9-0000-714b-e31d00000000" qualifiedName="quot;Oracle Data Warehousequot;quot;Catalogquot;
   .quot;dboquot;quot;Dim_W_PARTY_PER_D_Contactquot;"/>
</SamplingTable>
```

10.1.7 XML Elements: Seg Catalog

The seg catalog element corresponds to the segmentation catalog object in the Marketing metadata. Segmentation catalogs are Oracle BI subject areas that are enabled for segmentation. Segmentation catalogs provides a set of dimensions and fact measures that can be used to create segment criteria.

The seg catalog tag includes the following child elements:

Table 10–10 Seg Catalog Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefPresentationCatalog</td>
<td>References the subject area for this segmentation catalog.</td>
</tr>
<tr>
<td>QualifyingKeys</td>
<td>References the set of qualifying keys for this list catalog. Each qualifying key is contained in a sub-element called RefQualifyingKey.</td>
</tr>
</tbody>
</table>

The seg catalog tag does not include any attributes.

Syntax

```xml
<SegCatalog _>
   <Description />
   <RefPresentationCatalog _/>
   <QualifyingKeys>
      <RefQualifyingKey _/>
   </QualifyingKeys>
</SegCatalog>
```

Example

```xml
<SegCatalog name="Customers_A - Sample Sales"
   parentName="quot;Customersquot;"*
```
10.2 XML Security Elements

Security elements are typically those elements that are related to Oracle Business Intelligence security features, such as those related to variables, authentication, privileges, users, and application roles. Security elements include:

- Section 10.2.1, "XML Elements: Variable"
- Section 10.2.2, "XML Elements: LDAP Server"
- Section 10.2.3, "XML Elements: Init Block"
- Section 10.2.4, "XML Elements: Custom Authenticator"
- Section 10.2.5, "XML Elements: Privilege Package"
- Section 10.2.6, "XML Elements: Object Privilege"
- Section 10.2.7, "XML Elements: Query Privilege"
- Section 10.2.8, "XML Elements: User"
- Section 10.2.9, "XML Elements: Group"
- Section 10.2.10, "XML Elements: DB Sign-On"
- Section 10.2.11, "XML Elements: SetPermission"
- Section 10.2.12, "XML Elements: FMW Security"

10.2.1 XML Elements: Variable

The variable element corresponds to the variable object in the Physical layer of the repository.

The variable tag includes the following attributes:
The variable tag includes the following child elements:

| Table 10–12  Variable Child Elements |
|-----------------|--------------------------|
| **Element Name** | **Description** |
| Expr             | Contains the expression that forms the definition of this variable. |

**Syntax**

```xml
<Variable _
    securityMask=""
    isSessionVar="">
<Description />
<Expr>...</Expr>
</Variable>
```

**Example**

```xml
<Variable name='DYNAMIC_PRODUCT_TABLE'
    securityMask="" parentName="&quot;DUAL Prod Table&quot;" parentUid="80cc200a-0bd9-0000-714b-e31d00000000" parentId='3033:81463' id='3031:7079' uid='80caaa443-0bd7-0000-714b-e31d00000000'
    isSessionVar="true">
<Expr>
<![CDATA['SAMP_PRODUCTS_D']]> </Expr>
</Variable>
```

### 10.2.2 XML Elements: LDAP Server

The LDAP server element corresponds to the LDAP server object in the Identity Manager dialog in the Administration Tool.
The LDAP server tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostName</td>
<td>The host name of the LDAP server.</td>
</tr>
<tr>
<td>port</td>
<td>The port number of the LDAP server (for example, 389).</td>
</tr>
<tr>
<td>baseDN</td>
<td>The base distinguished name (DN) that identifies the starting point of the authentication search.</td>
</tr>
<tr>
<td>bindDN</td>
<td>The optional DN that is required to bind to the LDAP server. This option is required if isADSI is set to true.</td>
</tr>
<tr>
<td>userNameAttrType</td>
<td>A value that uniquely identifies a user, such as the user ID, the RDN (relative distinguished name), or (for ADSI) sAMAccountName.</td>
</tr>
<tr>
<td>domainID</td>
<td>A domain identifier that uniquely identifies the domain for which the LDAP object is responsible. This is especially useful when there are multiple LDAP objects.</td>
</tr>
<tr>
<td>password</td>
<td>The user password associated with the Bind DN. This option is required if isADSI is set to true. Note that passwords are encrypted.</td>
</tr>
<tr>
<td>timeout</td>
<td>The time period after which Administration Tool import attempts or Oracle BI Server user authentication attempts against this LDAP server will time out.</td>
</tr>
<tr>
<td>cacheExpiry</td>
<td>The interval at which the authentication cache entry in the LDAP cache for a logged on user is refreshed. This option corresponds to Cache refresh interval in the Repository tab of the Options dialog in the Administration Tool.</td>
</tr>
<tr>
<td>isADSI</td>
<td>If set to true, indicates that this is an Active Directory LDAP server.</td>
</tr>
<tr>
<td>useSSL</td>
<td>If set to true, indicates that the connection to the LDAP server uses Secure Sockets Layer (SSL).</td>
</tr>
<tr>
<td>keyFile</td>
<td>The name of the key file that holds the client certificate and Certificate Authority (CA) certificate. Required if useSSL is set to true. This option corresponds to Key file name in the Repository tab of the Options dialog in the Administration Tool.</td>
</tr>
<tr>
<td>keyPassword</td>
<td>For a secure LDAP, there is a keydb file which stores the required certificates. keyPassword is the password to access the keydb file.</td>
</tr>
<tr>
<td>version</td>
<td>The LDAP version number of the LDAP server, either 2 or 3. The default is 3.</td>
</tr>
<tr>
<td>cacheEntries</td>
<td>The maximum number of entries in the authentication cache, preallocated when the Oracle BI Server starts. If the number of users exceeds this limit, cache entries are replaced using the LRU algorithm. If this value is 0, then the authentication cache is disabled. This option corresponds to Number of Cache Entries in the Repository tab of the Options dialog in the Administration Tool.</td>
</tr>
</tbody>
</table>

The LDAP server tag does not include any child elements.
XML Security Elements

**Syntax**

```xml
<LDAPServer _
    hostName=""
    port=""
    baseDN=""
    bindDN=""
    userNameAttrType=""
    domainID=""
    password=""
    timeout=""
    cacheExpiry=""
    entities=""
    isADSI=""
    useSSL=""
    keyFile=""
    keyPassword=""
    version=""
    cacheEntries="">
    <Description />
</LDAPServer>
```

**Example**

```xml
<LDAPServer name="L1"
    id="3035:44085" uid="c39e1310-0c00-1000-806e-0a9702490000"
    hostName="sdacns05" port="389" version="3" baseDN="name1" bindDN="name2"
    userNameAttrType="sAMAccountName"
    password="6C6F0BE01FC4111F3AC2236A364D912E6CC6E805A7729C8CC34915F82D6CA5D1B19
        AF813BD5AD908750471EBF5D151A A886F77E060E2E17F65ADB8D18CAB2E4D3EFA15B75E30D8
        B4BFA8C7E2D705525D" timeout="60" cacheExpiry="86400" cacheEntries="1000"
    isADSI="true" useSSL="true"
    keyPassword="E3130008E1C4CAD47041E4AE68B048B6 7C2E35213306F12832914CBE7A90D95
        561D771DDE06484112B1FC6F27B6058">
    <Description />
</LDAPServer>
```

### 10.2.3 XML Elements: Init Block

The init block element corresponds to the initialization block object in the Physical layer. Initialization blocks are used to initialize dynamic repository variables, system session variables, and nonsystem session variables.

The init block tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isSessionVar</td>
<td>If set to true, indicates that this initialization block is a session initialization block. Session initialization blocks are used with session variables. The default value is false.</td>
</tr>
<tr>
<td>refreshPeriod</td>
<td>The refresh interval for this initialization block, in seconds. For repository initialization blocks only. The default value is 0.</td>
</tr>
<tr>
<td>baseTime</td>
<td>The start date and time for this initialization block. For repository initialization blocks only. The default value is 2001-01-01 00:00:00.</td>
</tr>
</tbody>
</table>
The init block tag includes the following child elements:

### Table 10–15  Init Block Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefConnectionPool</td>
<td>References the connection pool for this initialization block. This element is only used if the data source type for this initialization block is Database or XML.</td>
</tr>
<tr>
<td>RefCustomAuthenticator</td>
<td>References the custom authenticator for this initialization block.</td>
</tr>
</tbody>
</table>
| DBMap                      | Stores the value of database-specific strings. The `<DBMap>` element contains an `<Item>` tag, which in turn contains a `<Value>` tag, as follows:  
  `<Item name="">  
    <Value />
  </Item>` |
### Table 10–15 (Cont.) Init Block Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predecessors</td>
<td>References the initialization blocks, if any, that are initialized before this initialization block. Each initialization block is contained in a sub-element called RefInitBlock.</td>
</tr>
<tr>
<td>Variables</td>
<td>References the variables that are associated with this initialization block. Each variable is contained in a sub-element called RefVariable.</td>
</tr>
<tr>
<td>InitString</td>
<td>The SQL initialization string needed to populate the variables. This element is only used if the data source type for this initialization block is Database or XML.</td>
</tr>
<tr>
<td>LDAPServers</td>
<td>References the LDAP servers that are associated with this initialization block. Each LDAP server is contained in a sub-element called RefLDAPServer. This element is only used if the data source type for this initialization block is LDAP.</td>
</tr>
</tbody>
</table>

### Syntax

```
<InitBlock _
   isSessionVar=""
   refreshPeriod=""
   baseTime=""
   isRowWiseInit=""
   isRowWiseCacheEnabled=""
   isSuccRequired=""
   isEnabled=""
   isDeferredExecution=""
   isDBIndependentSQL="">
   <Description />
   <RefConnectionPool _/>
   <RefCustomAuthenticator _/>
   <DBMap>
     <Item name="">
       <Value _/>
     </Item>
   </DBMap>
   <Predecessors>
     <RefInitBlock _/>
   </Predecessors>
   <Variables>
     <RefVariable _/>
   </Variables>
   <InitString _/>
   <LDAPServers>
     <RefLDAPServer _/>
     <RefLDAPServer _/>
   </LDAPServers>
</InitBlock>
```

### Example

```
<InitBlock name="DUAL Prod Table"
           id="3033:81463" uid="80cc200a-0bd9-0000-714b-e31d00000000"
10.2.4 XML Elements: Custom Authenticator

The Custom Authenticator element corresponds to the Custom Authenticator object in the Identity Manager dialog in the Administration Tool.

The Custom Authenticator tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacheExpiry</td>
<td>The interval at which the authentication cache entry for a logged on user is refreshed for this custom authenticator.</td>
</tr>
<tr>
<td>path</td>
<td>The path and name of the authenticator plug-in dll.</td>
</tr>
<tr>
<td>configParam</td>
<td>Lists any parameters for this custom authenticator that have been explicitly exposed for configuration.</td>
</tr>
<tr>
<td>hiddenParam</td>
<td>Lists any parameters for this custom authenticator that have been encrypted, such as passwords.</td>
</tr>
<tr>
<td>cacheEntries</td>
<td>The maximum number of entries in the authentication cache for this custom authenticator, preallocated when the Oracle BI Server starts. If the number of users exceeds this limit, cache entries are replaced using the LRU algorithm. If this value is 0, then the authentication cache is disabled.</td>
</tr>
<tr>
<td>csfKey</td>
<td>This attribute is not used.</td>
</tr>
</tbody>
</table>

The Custom Authenticator element does not include any child elements.

Syntax

```xml
<CustomAuthenticator _
    cacheExpiry=""
    path=""
    configParam=""
    hiddenParam=""
    cacheEntries="">
    <Description />
</CustomAuthenticator>
```
Example

```xml
<CustomAuthenticator name="NEWCA"
  id="3047:44086" uid="c467a4a0-0c00-1000-806e-0a9702490000"
  cacheExpire="86400" cacheEntries="1000"
  path="authenticate.dll"
  configParam="ABC"
  hiddenParam="D7EDED84BC624A917F5B462A4DCA05CDCE256EBEEEDC97D5B383E1888393F5585DA9CAC25E1FA59">
</CustomAuthenticator>
```

10.2.5 XML Elements: Privilege Package

The privilege package element is a container for other privileges. It corresponds to a set of permissions that applies to a particular user or application role object in the repository metadata. The privilege package element contains the information available on the Object Permissions, Data Filters, and Query Limits tabs of the User/Application Role Permissions dialog in the Administration Tool.

Because privilege packages can be associated with more than one user or application role, the total number of privilege packages may not equal the total number of users and application roles.

**Note:** You can also set permissions for a privilege package without specifying values in the Administration Tool. For more information, see Section 10.2.11, "XML Elements: SetPermission."

The privilege package tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isFineGrainedMode</td>
<td>Reserved for Oracle internal use only.</td>
</tr>
</tbody>
</table>

The privilege package tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryPrivileges</td>
<td>References any query privileges that belong to this privilege package, along with the corresponding database.</td>
</tr>
<tr>
<td></td>
<td>Each pair of query privileges and database name is contained in a sub-element called Mapping.</td>
</tr>
<tr>
<td></td>
<td>Each Mapping sub-element contains two additional sub-elements: RefDatabase, which references the database name, and RefQueryPrivilege, which references the query privilege.</td>
</tr>
<tr>
<td></td>
<td>See the syntax for the privilege package element to see the complete hierarchy for the QueryPrivileges child element.</td>
</tr>
</tbody>
</table>
XML Security Elements

Note that the schema lists attributes for none, read, write, readwrite, delete, grant, and all, but these attributes are not used. Instead, these permissions are exposed as part of the child element definitions.

Syntax

```xml
<PrivilegePackage ... isFineGrainedMode=""/>
<Description />
<QueryPrivileges>
  <Mapping>
    <RefDatabase ...
      uid="0023840a-0bcf-0000-714b-e31d00000000" qualifiedName="Oracle Data Warehouse"/>
    <RefQueryPrivilege ...
      uid="808626e8-106c-0000-714b-e31d00000000" qualifiedName="QueryPriv_4204:3417391144981469"/>
  </Mapping>
</QueryPrivileges>
<Filters>
  <Mapping status="">
    <RefObject ...
      uid="808626e8-106c-0000-714b-e31d00000000" qualifiedName="QueryPriv_4204:3417391144981469"/>
  </Mapping>
</Filters>
<ObjectPrivileges>
  <RefObjectPrivilege />
</ObjectPrivileges>
</PrivilegePackage>
```

Example

```xml
<PrivilegePackage name="PrivPack_4209:3417381144981469"
  id="4209:281602" uid="808626e7-1071-0000-714b-e31d00000000">
  <QueryPrivileges>
    <Mapping>
      <RefDatabase id="3023:68561"
        uid="0023840a-0bcf-0000-714b-e31d00000000"
        qualifiedName="Oracle Data Warehouse"/>
      <RefQueryPrivilege id="4204:281350"
        uid="808626e8-106c-0000-714b-e31d00000000"
        qualifiedName="QueryPriv_4204:3417391144981469"/>
    </Mapping>
  </QueryPrivileges>
  <Filters>
    <Mapping status="">
      <RefObject ...
        uid="808626e8-106c-0000-714b-e31d00000000" qualifiedName="QueryPriv_4204:3417391144981469"/>
    </Mapping>
  </Filters>
</PrivilegePackage>
```

### Table 10–18 (Cont.) Privilege Package Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filters</td>
<td>References any data filters that belong to this privilege package by mapping an object name with the filter expression. Each pair of object name and filter expression is contained in a sub-element called Mapping. This sub-element has one attribute called Status. Valid values are Enabled, Disabled, and Ignored. Each Mapping sub-element contains two additional sub-elements: RefObject, which references the metadata object name, and Expr, which contains the filter expression. See the syntax for the privilege package element to see the complete hierarchy for the Filters child element.</td>
</tr>
<tr>
<td>ObjectPrivileges</td>
<td>References any object privileges that belong to this privilege package. Each object privilege is contained in a sub-element called RefObjectPrivilege.</td>
</tr>
</tbody>
</table>

Note that the schema lists attributes for none, read, write, readwrite, delete, grant, and all, but these attributes are not used. Instead, these permissions are exposed as part of the child element definitions.
10.2.6 XML Elements: Object Privilege

The object privilege element corresponds to the set of object permissions you can apply to metadata objects such as logical or presentation columns. In the Administration Tool, you can set object permissions in the Presentation layer, or in the Object Permissions tab of the User/Application Role Permissions dialog.

**Note:** You can also set permissions for an object privilege without specifying values in the Administration Tool. For more information, see Section 10.2.11, "XML Elements: SetPermission."

The object privilege tag includes the following attributes:
The object privilege tag includes the following child elements:

**Table 10–19 Object Privilege Tag Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| type           | Contains the type of privilege applied to the objects. Valid values are:  
|                | - read: corresponds to the Read privilege in the Administration Tool  
|                | - readWrite: corresponds to the Read/Write privilege in the Administration Tool  
|                | - none: corresponds to the No Access privilege in the Administration Tool  
|                | This attribute can contain the following other values: delete, write, grant, and all. These other values are for internal use. |

**Table 10–20 Object Privilege Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>References the set of objects for which this privilege type applies. Each object is contained in a sub-element called RefObject.</td>
</tr>
</tbody>
</table>

**Syntax**

```
<ObjectPrivilege type="">
  <Description />
  <Objects>
    <RefObject />
    <RefObject />
  </Objects>
</ObjectPrivilege>
```

**Example**

```
<ObjectPrivilege name='ObjPriv_1500:64621257864270'
  id='1500:6462' uid='80caalbb-05dc-0000-714b-e31d00000000'
  type='read'>
  <Objects>
    <RefObject id='4008:43061'
      uid='80cb6a2f-0fa8-0000-714b-e31d00000000'
      qualifiedName='&quot;Sample Sales&quot;&quot;Time&quot;'/>
    <RefObject id='4008:43085'
      uid='80cb6a47-0fa8-0000-714b-e31d00000000'
      qualifiedName='&quot;Sample Sales&quot;&quot;More Time Objects&quot;'/>
    <RefObject id='4008:43103'
      uid='80cb6a59-0fa8-0000-714b-e31d00000000'
      qualifiedName='&quot;Sample Sales&quot;&quot;Products&quot;'/>
  </Objects>
</ObjectPrivilege>
```

**10.2.7 XML Elements: Query Privilege**

The query privilege element corresponds to the set of query privileges you can apply to users or application roles. In the Administration Tool, you can set query privileges in the Query Limits tab of the User/Application Role Permissions dialog.
The query privilege tag includes the following attributes:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxExecTime</td>
<td>The maximum number of minutes queries can run on a database.</td>
</tr>
<tr>
<td>maxExecTimePrivilege</td>
<td>Specifies the status of the maximum query run time limit. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>Enable: Limits the time to the value specified in maxExecTime.</td>
</tr>
<tr>
<td></td>
<td>Disable: Disables any limits set in maxExecTime.</td>
</tr>
<tr>
<td></td>
<td>Warn: Does not enforce limits, but logs queries that exceed the set limit in the Query log.</td>
</tr>
<tr>
<td></td>
<td>Ignore: Inherits limits from the parent application role. If there is no time limit to inherit, no limit is enforced.</td>
</tr>
<tr>
<td>maxRows</td>
<td>The maximum number of rows for users to retrieve from a database.</td>
</tr>
<tr>
<td>maxRowsPrivilege</td>
<td>Specifies the status of the maximum number of rows limit. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>Enable: Limits the number of rows to the value specified. If the number of rows exceeds the maxRows value, the query is terminated.</td>
</tr>
<tr>
<td></td>
<td>Disable: Disables any limits set in maxRows.</td>
</tr>
<tr>
<td></td>
<td>Warn: Does not enforce limits, but logs queries that exceed the set limit in the Query log.</td>
</tr>
<tr>
<td></td>
<td>Ignore: Inherits limits from the parent application role. If there is no row limit to inherit, no limit is enforced.</td>
</tr>
<tr>
<td>populatePrivilege</td>
<td>Specifies whether the Populate privilege is granted or denied for a database. The Populate stored procedure writes the Cache/Saved Result Set value to the database when a criteria block is cached. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>Allow: Explicitly grants the Populate privilege.</td>
</tr>
<tr>
<td></td>
<td>Disallow: Explicitly denies the Populate privilege.</td>
</tr>
<tr>
<td></td>
<td>Ignore: Inherits limits from the parent application role. If there is no limit to inherit, then the Populate privilege is allowed or disallowed based on the property Allow populate queries by default for the database object.</td>
</tr>
<tr>
<td>execPhysicalPrivilege</td>
<td>Specifies whether direct database requests can be executed by default. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>Allow: Explicitly grants the ability to execute direct database requests.</td>
</tr>
<tr>
<td></td>
<td>Disallow: Explicitly denies the ability to execute direct database requests.</td>
</tr>
<tr>
<td></td>
<td>Ignore: Inherits limits from the parent application role. If there is no limit to inherit, then direct database requests are allowed or disallowed based on the property Allow direct database requests by default for the database object.</td>
</tr>
</tbody>
</table>

The query privilege tag includes the following child elements:
Table 10–22  Query Privilege Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecTimeTable</td>
<td>If database access has been restricted to particular time periods, lists</td>
</tr>
<tr>
<td></td>
<td>the time periods and whether access is allowed or disallowed. Each time</td>
</tr>
<tr>
<td></td>
<td>period is contained in a sub-element called Item. The Item child element has</td>
</tr>
<tr>
<td></td>
<td>three attributes: allow (true or false), startTime, and endTime.</td>
</tr>
</tbody>
</table>

Syntax

```xml
<QueryPrivilege _
    maxExecTime=""
    maxExecTimePrivilege=""
    maxRows=""
    maxRowsPrivilege=""
    populatePrivilege=""
    execPhysicalPrivilege="">
    <Description />
    <ExecTimeTable>
        <Item allow="" startTime="" endTime=""/>
        <Item allow="" startTime="" endTime=""/>
    </ExecTimeTable>
</QueryPrivilege>
```

Example

```xml
<QueryPrivilege name="QueryPriv_4204:275811265643135"
    id="4204:27581" uid="80cb1ael-106c-0000-714b-e31d00000000"
    maxExecTime="600" maxRows="100000"
    populatePrivilege="enable" execPhysicalPrivilege="enable">
</QueryPrivilege>
```

10.2.8 XML Elements: User

The user element corresponds to the user object in the repository. These objects represent the metadata repository copies or versions of externally managed user objects.

The user tag includes the following attributes:

Table 10–23  User Tag Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fullName</td>
<td>The full name of this user.</td>
</tr>
<tr>
<td>loggingLevel</td>
<td>The logging level for this user.</td>
</tr>
<tr>
<td>logStatisticsStatus</td>
<td>This attribute is not used.</td>
</tr>
<tr>
<td>guid</td>
<td>Global unique identifier for a given user, which is independent of the</td>
</tr>
<tr>
<td></td>
<td>user name.</td>
</tr>
</tbody>
</table>

The user tag includes the following child elements:

Table 10–24  User Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefPrivilegePackage</td>
<td>References the privilege package, or set of object privileges, data filters,</td>
</tr>
<tr>
<td></td>
<td>and query privileges, that apply to this user.</td>
</tr>
</tbody>
</table>
10.2.9 XML Elements: Group

The group element corresponds to the application role object in the repository metadata.

The group tag includes the following attributes:

**Groups**

References the application roles to which this user belongs. Each application role is contained in a sub-element called RefGroup.

**DBSignOnList**

References database login user names and passwords, along with the corresponding database, if this user has database-specific logins defined. Each pair of database credentials and database name is contained in a sub-element called Mapping. Each Mapping sub-element contains two additional sub-elements: RefDBSignOn, which references the database credentials, and RefDatabase, which references the database name. Note that passwords are encrypted. See the syntax for the user element to see the complete hierarchy for the DBSignOnList child element.

### Table 10–24 (Cont.) User Child Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>References the application roles to which this user belongs. Each application role is contained in a sub-element called RefGroup.</td>
</tr>
<tr>
<td>DBSignOnList</td>
<td>References database login user names and passwords, along with the corresponding database, if this user has database-specific logins defined. Each pair of database credentials and database name is contained in a sub-element called Mapping. Each Mapping sub-element contains two additional sub-elements: RefDBSignOn, which references the database credentials, and RefDatabase, which references the database name. Note that passwords are encrypted. See the syntax for the user element to see the complete hierarchy for the DBSignOnList child element.</td>
</tr>
</tbody>
</table>

### Syntax

```xml
<User _
    fullName="" 
    loggingLevel="" 
    guid="">
    <Description /> 
    <RefPrivilegePackage _/> 
    <Groups> 
      <RefGroup _/> 
      <RefGroup _/> 
    </Groups> 
    <DBSignOnList> 
      <Mapping> 
        <RefDBSignOn _/> 
        <RefDatabase _/> 
      </Mapping> 
    </DBSignOnList> 
</User>
```

### Example

```xml
<User name='mgr' id='1501:577531' 
    uid='80ca34ae-05dd-0000-714b-e31d00000000' 
    guid='D289B2F1E54411DEBFF92BEEA2576FF2'> 
    <RefPrivilegePackage _id='4209:588476' 
        uid='80d92bbd-1071-0000-714b-e31d00000000' 
        qualifiedName='"PrivPack_4209:5884761253725358"'/> 
    <Groups> 
      <RefGroup _id='4206:577530' 
        uid='80ca34b7-106e-0000-714b-e31d00000000' 
        qualifiedName='"Mgr"'/> 
    </Groups> 
</User>
```
The group tag includes the following child elements:

**Table 10–26  Group Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefPrivilegePackage</td>
<td>References the privilege package, or set of object privileges, data filters, and query privileges, that apply to this application role.</td>
</tr>
<tr>
<td>ChildGroups</td>
<td>References the set of application roles that belong to this application role. Each child application role is contained in a sub-element called RefGroup.</td>
</tr>
</tbody>
</table>

**Syntax**

```xml
<Group ...
   displayName=""
   guid="">
   <Description />
   <RefPrivilegePackage ...
   <ChildGroups>
     <RefGroup ...
   </ChildGroups>
</Group>
```

**Example**

```xml
<Group name="Marketing Analytics User" id="4206:10"
   uid="0023e1da-106e-0000-714b-e31d00000000"
   guid="Marketing Analytics User">
   <RefPrivilegePackage id="4209:281517"
     uid="0023e206-1071-0000-714b-e31d00000000"
     qualifiedName="&quot;PrivPack_4209:651191007624440&quot;"/>
   <ChildGroups>
     <RefGroup id="4206:26"
       uid="0023e1b5-106e-0000-714b-e31d00000000"
       qualifiedName="&quot;CM Marketing Analytics User&quot;"/>
   </ChildGroups>
</Group>
```

### 10.2.10 XML Elements: DB Sign-On

The DB sign-on element corresponds to the information provided in the Logons tab of the user object. If the repository has been set up for database-specific login IDs and passwords, you can set up database-specific credentials to allow privileged users to connect to the underlying databases at the level of authority granted to those users in the databases.

The DB sign-on tag includes the following attributes:
The DB sign-on tag does not include any child elements.

### Syntax

```xml
<DBSignOn ... user=''
password=''>
<Description />
</DBSignOn>
```

### Example

```xml
<DBSignOn name='LOGON_4211:2559932074113'
id='4211:158' uid='309a5c29-0bf7-1000-8f7c-0ae593260000'
user='my_db_user'
password='BA0EAAB163'>
</DBSignOn>
```

10.2.11 XML Elements: SetPermission

The SetPermission element is a special element, in that it does not represent a repository object. Its purpose is to define permissions to be automatically added to the Object Privilege and Privilege Package objects, without the requirement to explicitly define those objects. For more information on the objects for which these permissions can be defined, see Section 10.2.5, "XML Elements: Privilege Package" and Section 10.2.6, "XML Elements: Object Privilege."

The SetPermission element is effectively a command that is designed to set permissions on objects to users and roles.

Depending on the sub-elements chosen, SetPermission can be used to add or suppress user and role permissions on a variety of presentation and marketing objects, namely:

- Connection Pool
- List Catalog
- Presentation Catalog
- Presentation Column
- Presentation Hierarchy
- Presentation Level
- Presentation Table
- Target Level

The SetPermission element, with selected sub-element and sub-element values, is first added to the XML file which is input to the XML command-line tools biserverxmlexec and biserverxmlcli. For more information on the command-line tools, see Section 6.2, "Generating and Executing XML."

When those tools are executed, SetPermission sets the appropriate permissions in the Object Privilege and Privilege Package objects that are generated automatically in

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>The user name for the underlying database.</td>
</tr>
<tr>
<td>password</td>
<td>The corresponding password for the underlying database. Note that passwords are encrypted.</td>
</tr>
</tbody>
</table>
output RPD. If these Object Privilege and Privilege Package objects already exist in the
RPD, they will be overwritten by the execution of the SetPermission command.

---

**Note:** The SetPermission element is only used to add information into a repository; it will not appear in the output XML file that can be generated from the Administration Tool for a repository.

---

The minimum requirement for a SetPermission element is that it must consist of at least one object and one role or user. The objects that are specified can exist in either in the XML input file or the RPD or both. The roles and users must be in the security realm and must also be present in the RPD or XML input file or both.

The SetPermission tag does not include any attributes.

The SetPermission tag includes the following child elements:

**Table 10–28 SetPermission Child Elements**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>References the set of objects for which this SetPermission applies. Each object is contained in one of the following sub-elements: RefPresentationCatalog, RefPresentationTable, RefPresentationColumn, RefPresentationHierarchy, RefPresentationLevel, RefTargetLevel, RefListCatalog, each of which references its associated object type.</td>
</tr>
<tr>
<td>Role</td>
<td>Contains the attributes name and permission. The permission values can be read, write, none, or default.</td>
</tr>
<tr>
<td>User</td>
<td>Contains the attributes name and permission. The permission values can be read, write, none, or default.</td>
</tr>
</tbody>
</table>

**Syntax**

```
<SetPermission>
   <Objects>
      <RefPresentationCatalog.../>
      <RefPresentationTable.../>
      <RefPresentationColumn.../>
      <RefPresentationHierarchy.../>
      <RefPresentationLevel.../>
      <RefTargetLevel.../>
      <RefListCatalog.../>
   </Objects>
   <Role name="" permission=""/>
   <User name="" permission=""/>
</SetPermission>
```

**Example**

```
<SetPermission>
   <Objects>
      <RefPresentationCatalog
         qualifiedName="&quot;custExt3c2ce61c_0dc2_420c_b785_cf6b451deda5&quot;" />
   </Objects>
   <Role name="SALES_EXECUTIVE_TRANSACTION_ANALYSIS_DUTY" permission="none"/>
   <Role name="SALES_CAMPAIGN_TRANSACTION_ANALYSIS_DUTY" permission="read"/>
   <Role name="SALES_MANAGERIAL_TRANSACTION_ANALYSIS_DUTY" permission="read"/>
   <Role name="SALES_TRANSACTION_ANALYSIS_DUTY" permission="read"/>
</SetPermission>
```
10.2.12 XML Elements: FMW Security

The FMW security element is not used.

10.3 Other XML Elements

The elements described in this section are miscellaneous elements that do not fit into any other category. These elements include:

- Section 10.3.1, "XML Elements: Project"
- Section 10.3.2, "XML Elements: Query Object"

10.3.1 XML Elements: Project

The project element corresponds to the project object in the repository metadata. Projects support multiuser development by enabling developers to work on projects in their area of responsibility.

The project tag includes the following child elements:

<table>
<thead>
<tr>
<th>Table 10–29</th>
<th>Project Child Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Catalogs</td>
<td>References the subject areas that belong to the project. Each subject area is contained in a sub-element called RefPresentationCatalog.</td>
</tr>
<tr>
<td>FactTables</td>
<td>References the logical fact tables that belong to the project. Each fact table is contained in a sub-element called RefLogicalTable.</td>
</tr>
<tr>
<td>Groups</td>
<td>References the application roles that belong to the project. Each application role is contained in a sub-element called RefGroup.</td>
</tr>
<tr>
<td>Users</td>
<td>References the users that belong to the project. Each user is contained in a sub-element called RefUser.</td>
</tr>
<tr>
<td>Variables</td>
<td>References the variables that belong to the project. Each variable is contained in a sub-element called RefVariable.</td>
</tr>
<tr>
<td>InitBlocks</td>
<td>References the initialization blocks that belong to the project. Each initialization block is contained in a sub-element called RefInitBlock.</td>
</tr>
<tr>
<td>ListCatalogs</td>
<td>References the list catalogs that belong to the project. Each list catalog is contained in a sub-element called RefListCatalog.</td>
</tr>
<tr>
<td>Tables</td>
<td>References the presentation tables that belong to the project. Each presentation table is contained in a sub-element called RefPresentationTable.</td>
</tr>
<tr>
<td>Columns</td>
<td>References the presentation columns that belong to the project. Each presentation column is contained in a sub-element called RefPresentationColumn.</td>
</tr>
</tbody>
</table>
The project tag does not include any attributes.

**Syntax**

```xml
<Project _>
    <Description />
    <Catalogs>
        <RefPresentationCatalog _/>
    </Catalogs>
    <FactTables>
        <RefLogicalTable _/>
    </FactTables>
    <Groups>
        <RefGroup _/>
    </Groups>
    <Users>
        <RefUser _/>
    </Users>
    <Variables>
        <RefVariable _/>
    </Variables>
</Project>
```
Example

<Project name="FUSION_SPE"
    id="4213:281691" uid="80ca6265-1075-0000-714b-e31d00000000">
    <Catalogs>
        <RefPresentationCatalog id="4004:275486"
            uid="80ca99dbb-0fa4-0000-714b-e31d00000000"
            qualifiedName="Sales Manager Dashboard - Campaign"/>
        <RefPresentationCatalog id="4004:275495"
            uid="80ca9e05-0fa4-0000-714b-e31d00000000"
            qualifiedName="Sales Manager Dashboard - Lead"/>
    </Catalogs>
    <FactTables>
        <RefLogicalTable id="2035:9160"
            uid="00224fb6-07f3-0000-714b-e31d00000000"
            qualifiedName="Core".Fact - CRM - Order Item"/>
        <RefLogicalTable id="2035:9371"
            uid="00224fb8-07f3-0000-714b-e31d00000000"
            qualifiedName="Core".Fact - CRM - Service Request"/>
    </FactTables>
    <Groups>
        <RefGroup id="4206:1"
            uid="0023e1a2-106e-0000-714b-e31d00000000"
Other XML Elements

```xml
<Groups>
  <Users>
    <RefUser id="1501:2"
      uid="0021f847-05dd-0000-714b-e31d00000000"
      qualifiedName="&quot;Administrator&quot;"/>
  </Users>
  <Variables>
    <RefVariable id="3031:250"
      uid="808466f1-0bd7-0000-714b-e31d00000000"
      qualifiedName="&quot;User Hierarchy Level&quot;.&quot;HIER_LEVEL&quot;"/>
    <RefVariable id="3031:337"
      uid="80c96155-0bd7-0000-714b-e31d00000000"
      qualifiedName="&quot;GetCRMCalendar&quot;.&quot;CRM_CAL_ID&quot;"/>
  </Variables>
  <InitBlocks>
    <RefInitBlock id="3033:68605"
      uid="808466f2-0bd9-0000-714b-e31d00000000"
      qualifiedName="&quot;User Hierarchy Level&quot;"/>
    <RefInitBlock id="3033:68645"
      uid="80c9615a-0bd9-0000-714b-e31d00000000"
      qualifiedName="&quot;GetCRMCalendar&quot;"/>
  </InitBlocks>
  <ExcludedColumns>
    <RefPresentationColumn id="4010:1181008"
      uid="80ec06eb-0faa-0000-714b-e31d00000000"
      qualifiedName="&quot;Sales Manager Dashboard - Customer&quot;.&quot;Date&quot;.&quot;.Day of Week&quot;"/>
    <RefPresentationColumn id="4010:1181009"
      uid="80ec06ec-0faa-0000-714b-e31d00000000"
      qualifiedName="&quot;Sales Manager Dashboard - Customer&quot;.&quot;Date&quot;.&quot;.Day of Year&quot;"/>
  </ExcludedColumns>
</Groups>
</Project>

10.3.2 XML Elements: Query Object

The query object element corresponds to the saved query object in the repository metadata. You can save a query from the Query Repository dialog in the Administration Tool.

The query object tag includes the following child elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryString</td>
<td>Contains the query string that qualifies and defines the query.</td>
</tr>
</tbody>
</table>

The query object tag does not include any attributes.

Syntax

```xml
<QueryObject ...>
  <Description />
  <QueryString .../>
</QueryObject>
```
Example

例

```xml
<QueryObject name="samplequery"
           id="4217:970" uid="c7fcd3b1-0c00-1000-8e0b-0a9702490000">
  <QueryString>
    <![CDATA[(128 = "***") AND (6 = Business Model) AND (0)]]>
  </QueryString>
</QueryObject>
```
This glossary defines terms for Oracle Business Intelligence Enterprise Edition. See also the Oracle Fusion Middleware Master Glossary for additional terms and definitions.

**action**
Provides functionality to navigate to related content or to invoke operations, functions or processes in external systems. You can include actions in analyses, dashboard pages, agents, scorecard objectives, scorecard initiatives, and KPIs.

See also **action link**.

**Action Framework**
The Action Framework is a component of the Oracle BI EE architecture and includes a J2EE application called the Action Execution Service (AES) and actions-specific JavaScript functionality deployed as part of Oracle BI EE. The action framework also includes client-side functionality for creating actions and invoking certain action types directly from the browser.

**action link**
A link to an action that you have embedded in an analysis, dashboard page, scorecard objective, scorecard initiative, or KPI that, when clicked, runs an associated action.

See also **action**.

**ADF Business Intelligence Component**
Provides the developer the ability to include Oracle BI Presentation Catalog objects in ADF Applications. This component uses a SOAP connection to access the catalog.

**Administration Server**
Part of the WebLogic server domain and runs the processes that manage Oracle Business Intelligence components. The Administration Server includes the Oracle WebLogic Server Administration Console, Oracle Fusion Middleware Control, and JMX MBeans. For a Simple Install type, the Administration Server also includes Java components for Oracle Business Intelligence such as Oracle BI Publisher and Oracle Real-Time Decisions.

See also **Fusion Middleware Control, Java components** and **Managed Server**.

**Administration Tool**
See **Oracle BI Administration Tool**.
advanced trellis
A trellis view that can display multiple visualization types within its grid, for example, Spark Line graphs, Spark Bar graphs, and numbers. Each visualization type displays a different measure.

You can think of an advanced trellis as a pivot table, except that for each measure you add to the pivot table, you can optionally associate a dimension and render that dimension as a spark graph visualization.

See also trellis and visualization.

agent
Enables you to automate your business processes. You can use agents to provide event-driven alerting, scheduled content publishing, and conditional event-driven action execution.

Agents can dynamically detect information-based problems and opportunities, determine the appropriate individuals to notify, and deliver information to them through a wide range of devices (email, phones, and so on).

aggregate persistence
A feature that automates the creation and loading of aggregate tables and their corresponding Oracle Business Intelligence metadata mappings to enable aggregate navigation.

aggregate table
A table that stores precomputed results from measures that have been aggregated over a set of dimensional attributes. Each aggregate table column contains data at a given set of levels. For example, a monthly sales table might contain a precomputed sum of the revenue for each product in each store during each month. Using aggregate tables optimizes performance.

aggregation rule
In an Oracle BI repository, a rule applied to a logical column or physical cube column that specifies a particular aggregation function to be applied to the column data, such as SUM.

In Presentation Services, users can see the rules that have been applied in the repository. Users can also change the default aggregation rules for measure columns.

alias table
A physical table that references a different physical table as its source. You can use alias tables to set up multiple tables, each with different keys, names, or joins, when a single physical table must serve in different roles. Because alias table names are included in physical SQL queries, you can also use alias tables to provide meaningful table names, making the SQL statements easier to read.

analysis
A query that a user creates on the Criteria tab in Presentation Services. An analysis can optionally contain one or more filters or selection steps to restrict the results.

See also filter and selection step.

analysis criteria
Consists of the columns, filters, and selection steps that you specify for an analysis.

See also analysis.
**analysis prompt**
A prompt that is added to an analysis. When the user selects a prompt value, that value then determines the content that displays in the analysis that contains the prompt, only.
See [dashboard prompt](#) and [prompt](#).

**attribute**
The details of a dimension in an Oracle BI repository. Attributes usually appear as columns of a dimension table.

**attribute column**
In Presentation Services, a column that holds a flat list of values that are also known as members. No hierarchical relationship exists between these members, as is the case for members of a hierarchical column. Examples include ProductID or City.
See [hierarchical column](#).

**BI Composer**
BI Composer is a simple-to-use wizard that enables you to quickly and easily create, edit, or view analyses without the complexities of the Analysis editor.

**BI domain**
Contains configurable system components (the coreapplication) and Java components (the WebLogic server domain), and includes the web-based management tools and applications that use resources.

A BI domain can be a set of middleware homes spread across one or more physical servers.
See also [BI instance](#).

**BI instance**
Refers to the system components (coreapplication) of a BI domain
See also [BI domain](#).

**BI object**
A piece of business intelligence content that is created with Presentation Services and saved to the Oracle BI Presentation Catalog. Examples of BI objects include analyses, dashboards, dashboard pages, scorecards, and KPIs.

**BI Search**
A search tool that resides outside of Presentation Services. BI Search is available from the Home Page after the administrator adds a link to the BI Search URL. BI Search provides a mechanism for searching for objects in the Oracle BI Presentation Catalog that is similar to a full-text search engine.

**bookmark link**
Captures the path to a dashboard page and all aspects of the page state.
See [prompted link](#).

**bridge table**
A table that enables you to resolve many-to-many relationships between two other tables.
briefing book
See Oracle BI Briefing Books.

business model
An object in the Oracle BI repository that contains the business model definitions and the mappings from logical to physical tables. Business models are always dimensional, unlike objects in the Physical layer, which reflect the organization of the data sources. Each business model contains logical tables, columns, and joins.

Business Model and Mapping layer
A layer of the Oracle BI repository that defines the business, or logical, model of the data and specifies the mapping between the business model and the Physical layer schemas. This layer can contain one or more business models.

The Business Model and Mapping layer determines the analytic behavior that is seen by users, and defines the superset of objects available to users. It also hides the complexity of the source data models.

business owner
The person responsible for managing and improving the business value and performance of a KPI or scorecard object, such as an objective, cause & effect map, and so on.

catalog
See Oracle BI Presentation Catalog.

cause & effect map
A component of a scorecard that lets you illustrate the cause and effect relationships of an objective or KPI.

See also Oracle Scorecard and Strategy Management.

chronological key
A column in a time dimension that identifies the chronological order of the members within a dimension level. The key must be unique at its level.

Cluster Controller
A process that serves as the first point of contact for new requests from Presentation Services and other clients. The Cluster Controller determines which Oracle BI Server in the cluster to direct the request to based on Oracle BI Server availability and load. It monitors the operation of servers in the cluster, including the Oracle BI Scheduler instances. The Cluster Controller is deployed in active-passive configuration.

column
In an Oracle BI repository, columns can be physical columns, logical columns, or presentation columns.

In Presentation Services, indicates the pieces of data that an analysis returns. Together with filters and selection steps, columns determine what analyses contain. Columns also have names that indicate the types of information that they contain, such as Account and Contact.

See also analysis, attribute column, hierarchical column, and measure column.
column filter
See filter.

column prompt
A type of filter that enables you to build specific value prompts on a data column to either exist alone on the dashboard or analysis or to expand or refine existing dashboard and analysis filters.
See also prompt.

complex join
A join in the Physical layer of an Oracle BI repository that uses an expression other than equals.

criterion
Objects that return a single Boolean value based on the evaluation of an analysis or of a key performance indicator (KPI). You use conditions to determine whether agents deliver their content and execute their actions, whether actions links are displayed in dashboard pages, or whether sections and their content are displayed in dashboard pages.
See also action, action link, agent and key performance indicator (KPI).

collection pool
An object in the Physical layer of an Oracle BI repository that contains the connection information for a data source.
See also Physical layer.

content designer
The user who creates business intelligence objects such as analyses, dashboards, and scorecards.

contextual event action
A predelivered action that uses the Action Framework to pass content from the business intelligence object to another region on an ADF page.
See also action, Action Framework, and action link.

criteria
See analysis criteria.

cube
An OLAP (online analytical processing) data structure that lets data be analyzed more quickly and with greater flexibility than structures in relational databases. Cubes are made up of measures and organized by dimensions. Cubes in multidimensional data sources roughly correspond to star schemas in relational database models.

currency prompt
A prompt that enables the user to change the currency type that displays in the currency columns on an analysis or dashboard.
See also prompt.
custom view
A component of a scorecard that lets you show a customized view of your business and strategy data.
See also Oracle Scorecard and Strategy Management.

dashboard
An object that provides personalized views of corporate and external information. A dashboard consists of one or more pages. Pages can display anything that you can access or open with a web browser, such as results of analyses, images, alerts from agents, and so on.
dashboard prompt
A prompt that is added to the dashboard. When the user selects a prompt value, that value then determines the content that displays in all analyses that are included on the dashboard.
See analysis prompt and prompt.

Dashboard URL
Used for incorporating or referencing the content of a specific dashboard in external portals or applications. It has several forms and optional arguments that you can use to control its behavior.

data source name (DSN)
A data structure that contains the information about a specific database, typically used by an ODBC driver to connect to the database. The DSN contains information such as the name, directory, and driver of the database.
Connection pool objects in the Physical layer of the Oracle BI repository contain DSN information for individual data sources.

database hint
Instructions placed within a SQL statement that tell the database query optimizer the most efficient way to execute the statement. Hints override the optimizer’s execution plan, so you can use hints to improve performance by forcing the optimizer to use a more efficient plan. Hints are supported only for Oracle Database data sources.

dimension
A hierarchical organization of logical columns (attributes). One or more logical dimension tables might be associated with at most one dimension.
A dimension might contain one or more (unnamed) hierarchies. There are two types of logical dimensions: dimensions with level-based hierarchies (structure hierarchies), and dimensions with parent-child hierarchies (value hierarchies).
A particular type of level-based dimension, called a time dimension, provides special functionality for modeling time series data.
See also hierarchy.

dimension table
A logical table that contains columns used by a particular dimension. A dimension table cannot be a fact table.
See also fact table.
driving table
A mechanism used to optimize the manner in which the Oracle BI Server processes multi-database joins when one table is very small (the driving table) and the other table is very large.

DSN
See data source name (DSN).

event polling table
Event polling tables (also called event tables) provide information to the Oracle BI Server about which physical tables have been updated. They are used to keep the query cache up-to-date. The Oracle BI Server cache system polls the event table, extracts the physical table information from the rows, and purges stale cache entries that reference those physical tables.

Essbase
A multidimensional database management system available from Oracle that provides a multidimensional database platform upon which to build business intelligence applications. Also referred to as Oracle's Hyperion Essbase.

fact table
In an Oracle BI repository, a logical table in the Business Model and Mapping layer that contains measures and has complex join relationships with dimension tables.
See also dimension table.

filter
Criteria that are applied to attribute and measure columns to limit the results that are displayed when an analysis is run. For measure columns, filters are applied before the query is aggregated and affect the query and thus the resulting values.
See also prompt and selection step.

focus node
The center circle of a strategy contribution wheel. For a strategy contribution wheel, the focus node represents the starting objective of the diagram.
See also strategy contribution wheel and focus trail.

focus trail
A series of chained circles that represent the node in the center of the strategy contribution wheel and any of its ancestors that are included in the diagram. Each circle displays the status color of its corresponding node.
See also strategy contribution wheel and focus node.

foreign key
A column or a set of columns in one table that references the primary key columns in another table.

fragmentation content
The portion, or fragment, of the set of data specified in a logical table source when the logical table source does not contain the entire set of data at a given level. Fragmentation content is defined by the logical columns that are entered in the Fragmentation content box in the Content tab of the Logical Table Source dialog box.
**Fusion Middleware Control**
Provides web-based management tools that enable you to monitor and configure Fusion Middleware components.

**global header**
An object in the user interface for Oracle BI Presentation Services that contains links and options that enable the user to quickly begin a task or locate a specific object within the Oracle BI Presentation Catalog. The global header always displays in the Presentation Services user interface, thus enabling users to quickly access links and search the catalog without having to navigate to the Home Page or Catalog page.

**Go URL**
Used to incorporate specific business intelligence results into external portals or applications. The Go URL is used when you add a result to your favorites or add a link to a request to a dashboard or external web site. It has several forms and optional arguments that you can use to control its behavior.

**hierarchical column**
In Presentation Services, a column that holds data values that are organized using both named levels and parent-child relationships. This column is displayed using a tree-like structure. Individual members are shown in an outline manner, with lower-level members rolling into higher-level members. For example, a specific day belongs to a particular month, which in turn is within a particular year. Examples include Time or Geography.

See also **attribute column**.

**hierarchy**
In an Oracle BI repository, a system of levels in a logical dimension that are related to each other by one-to-many relationships. All hierarchies must have a common leaf level and a common root (all) level.

Hierarchies are not modeled as separate objects in the metadata. Instead, they are an implicit part of dimension objects.

See also **dimension, logical level, and presentation hierarchy**.

**hierarchy level**
In Presentation Services, an object within a hierarchical column that either rolls up or is rolled up from other levels. Corresponds to a presentation level in an Oracle BI repository.

See also **presentation level**.

**home page**
Provides an intuitive, task-based entry way into the functionality of Presentation Services. The Home page is divided into sections that enable you to quickly begin specific tasks, locate an object, or access technical documentation.

**image prompt**
A prompt that provides an image with different areas mapped to specific values. The user clicks an image area to select the prompt value that populates the analysis or dashboard.

See also **prompt**.
**initialization block**
Used to initialize dynamic repository variables, system session variables, and nonsystem session variables. An initialization block contains the SQL statements that are executed to initialize or refresh the variables that are associated with that block.

**initiative**
Used in a scorecard, an initiative is a time-specific task or project that is necessary to achieve objectives. As such, you can use initiatives that support objectives as milestones as they reflect progress toward strategy targets.

See also objective and Oracle Scorecard and Strategy Management.

**inner graph**
A nested graph, inside the grid of a trellis graph. Each inner graph has its own dimensionality as specified in the Visualization area of the Layout pane.

See also trellis and outer edge.

**Java components**
Fusion Middleware Control components that are deployed as one or more Java EE applications (and a set of resources) and are managed by Node Manager.

See also Node Manager.

**key performance indicator (KPI)**
A measurement that defines and tracks specific business goals and strategic objectives. KPIs often times roll up into larger organizational strategies that require monitoring, improvement, and evaluation. KPIs have measurable values that usually vary with time, have targets to determine a score and performance status, include dimensions to allow for more precise analysis, and can be compared over time for trending purposes and to identify performance patterns.

See also Oracle Scorecard and Strategy Management.

**KPI watchlist**
A method of distributing KPIs to end users. A watchlist is a collection of KPIs that are built by adding the KPIs that are stored in the Oracle BI Presentation Catalog. After a KPI watchlist is built and saved, it is stored as a catalog object and can be added to dashboards and scorecards.

See also key performance indicator (KPI), watchlist, and Oracle Scorecard and Strategy Management.

**level**
See hierarchy level.

**logical display folder**
Folders used to organize objects in the Business Model and Mapping layer of an Oracle BI repository. They have no metadata meaning.

**logical join**
Joins that express relationships between logical tables. Logical joins are conceptual, rather than physical, joins. In other words, they do not join to particular keys or columns. A single logical join can correspond to many possible physical joins.
logical layer

See Business Model and Mapping layer.

logical level

In an Oracle BI repository, a component of a level-based hierarchy that either rolls up or is rolled up from other levels.

Parent-child hierarchies have implicit, inter-member levels between ancestors and descendants that are not exposed as logical level objects in the metadata. Although parent-child hierarchies also contain logical level objects, these levels are system generated and exist to enable aggregation across all members only.

See also dimension and hierarchy.

Logical SQL

The SQL statements that are understood by the Oracle BI Server. The Oracle BI Server Logical SQL includes standard SQL, plus special functions (SQL extensions) like AGO, TODATE, EVALUATE, and others.

Clients like Presentation Services send Logical SQL to the Oracle BI Server when a user makes a request. In addition, Logical SQL is used in the Business Model and Mapping layer to enable heterogeneous database access and portability. The Oracle BI Server transforms Logical SQL into physical SQL that can be understood by source databases.

logical table

A table object in the Business Model and Mapping layer of an Oracle BI repository. A single logical table can map to one or more physical tables. Logical tables can be either fact tables or dimension tables.

See also dimension table and fact table.

logical table source

Objects in the Business Model and Mapping layer of an Oracle BI repository that define the mappings from a single logical table to one or more physical tables. The physical to logical mapping can also be used to specify transformations that occur between the Physical layer and the Business Model and Mapping layer, and to enable aggregate navigation and fragmentation.

Managed Server

An individual J2EE application container (JMX MBean container). It provides local management functions on individual hosts for Java components and system components contained within the local middleware home, and refers to the Administration Server for all of its configuration and deployment information.

See also Administration Server and Fusion Middleware Control.

MDS

Oracle Metadata Services. A core technology of the Application Development Framework. MDS provides a unified architecture for defining and using metadata in an extensible and customizable manner.

See also MDS XML.

MDS XML

An XML format that is compatible with Oracle Metadata Services. MDS XML is a supported format for the Oracle BI repository. It enables integration with third-party source control management systems for offline repository development.
MDS XML format is different from the XML format generated by the Oracle BI Server XML API.

See also MDS, Oracle BI repository, and Oracle BI Server XML API.

**measure column**
A column that can change for each record and can be added up or aggregated. Typical measures are sales dollars and quantity ordered. Measures are calculated from data sources at query time.

Measure columns are displayed in the Oracle BI repository, usually in fact tables, or in Presentation Services.

**metadata**
Data about data. Metadata objects include the descriptions of schemas (such as tables, columns, data types, primary keys, foreign keys, and so on) and logical constructs (like fact tables, dimensions, and logical table source mappings).

The Oracle BI repository is made up of the metadata used by the Oracle BI Server to process queries.

**metadata dictionary**
A static set of XML documents that describe metadata objects, such as a column, including its properties and relationships with other metadata objects. A metadata dictionary can help users obtain more information about metrics or attributes for repository objects.

**microchart**
A tiny graph displayed in a grid along with other tiny graphs and numbers, comprising the data cell contents of an advanced trellis view. In Oracle BI EE, a microchart is always a spark graph.

See also advanced trellis and spark graph.

**mission statement**
A statement in a scorecard that specifies the key business goals and priorities that are required to achieve your vision.

See also vision statement and Oracle Scorecard and Strategy Management.

**multi-database join**
A join between two tables in an Oracle BI repository, where each table resides in a different database.

**Node Manager**
A daemon process that provides remote server start, stop, and restart capabilities when Java processes become unresponsive or terminate unexpectedly.

See also Java components.

**OCI**
See Oracle Call Interface (OCI).

**ODBC**
See Open Database Connectivity (ODBC).
**object properties**
Information about an object and attributes that the owner can assign to an object. Examples of properties include name, description, date stamps, read-only access, and do not index flag.
See also **permissions**.

**objective**
A required or desired outcome in a scorecard that forms your corporate strategy.
See also **initiative** and **Oracle Scorecard and Strategy Management**.

**offline mode**
In the Oracle BI Administration Tool, a mode where a repository builder can edit a repository that is not loaded into the Oracle BI Server.

**online mode**
In the Oracle BI Administration Tool, a mode where a repository builder can edit a repository while it is available for query operations. Online mode also allows user session monitoring for users connected to the subject areas in the repository.

**opaque view**
A Physical layer table that consists of a SELECT statement. In the Oracle BI repository, opaque views appear as view tables in the physical databases, but the view does not actually exist.

**Open Database Connectivity (ODBC)**
A standard interface used to access data in both relational and nonrelational databases. Database applications can use ODBC to access data stored in different types of database management systems, even if each database uses a different data storage format and programming interface.

**OPMN**
See **Oracle Process Manager and Notification Server (OPMN)**.

**Oracle BI Administration Tool**
A Windows application that is used to create and edit Oracle BI repositories. The Administration Tool provides a graphical representation of the three parts of a repository: the Physical layer, the Business Model and Mapping layer, and the Presentation layer.

**Oracle BI Briefing Books**
A collection of static or updatable snapshots of dashboard pages, individual analyses, and BI Publisher reports. You can download briefing books in PDF or MHTML format for printing and viewing. You also can update, schedule, and deliver briefing books using agents.

**Oracle BI JavaHost**
A service that gives Presentation Services the ability to use functionality that is provided in Java libraries to support components such as graphs. The services are provided based on a request-response model.
**Oracle BI Logical SQL View Object**

Provides the developer the ability to create a Logical SQL statement to access the Oracle BI Server and fetch business intelligence data and bind it to native ADF components for inclusion on an ADF page. This view object uses a BI JDBC connection to the Oracle BI Server.

**Oracle BI Presentation Catalog**

Stores business intelligence objects, such as analyses and dashboards, and provides an interface where users create, access, and manage objects, and perform specific object-based tasks (for example, export, print, and edit). The catalog is organized into folders that are either shared or personal.

**Oracle BI Presentation Services**

Provides the framework and interface for the presentation of business intelligence data to web clients. It maintains a Presentation Catalog service on the file system for the customization of this presentation framework. It is a standalone process and communicates with the Oracle BI Server using ODBC over TCP/IP. It consists of components that are known as Answers, Delivers, and Interactive Dashboards.

See also [ODBC](#); [Oracle BI Server](#); [Oracle BI Presentation Catalog](#); [Oracle BI Presentation Services](#).

**Oracle BI Presentation Services server**

The Oracle BI web server that exchanges information and data with the Oracle BI Server.

**Oracle BI Publisher**

A J2EE application that provides enterprise-wide publishing services in Oracle Business Intelligence. It generates highly formatted, pixel-perfect reports.

See also [report](#).

**Oracle BI Publisher report**

See [report](#).

**Oracle BI repository**

The set of Oracle Business Intelligence metadata that defines logical schemas, physical schemas, physical-to-logical mappings, aggregate table navigation, and other constructs. Oracle BI repositories can be in binary (RPD) format, in which repository metadata is contained in a single file with an extension of .rpd, or in a set of MDS XML documents. MDS XML format repositories are used for offline development only and cannot be loaded into the Oracle BI Server. Oracle BI repositories in both formats can be edited using the Oracle BI Administration Tool.

See also [metadata](#) and [Oracle BI Administration Tool](#).

**Oracle BI Scheduler**

An extensible scheduling application for scheduling results to be delivered to users at specified times. It is the engine behind the Oracle BI Delivers feature.

See also [results](#).

**Oracle BI Server**

A standalone process that maintains the logical data model that it provides to Presentation Services and other clients through ODBC. Metadata is maintained for the
data model in a local proprietary file called the repository file. The Oracle BI Server processes user requests and queries underlying data sources.

**Oracle BI Server XML API**

Provides utilities to create a generic, XML-based representation of the Oracle BI repository metadata. You can use this XML file version of the repository to programmatically modify the metadata. The Oracle BI Server XML API objects correspond to metadata repository objects in an RPD file. These objects differ from XML objects in the Oracle BI Presentation Catalog.

The XML generated by the Oracle BI Server XML API is different from the MDS XML format used for Oracle BI repositories integrated with third-party source control management systems.

See also [MDS XML](#).

**Oracle Business Intelligence Mobile**

Oracle Business Intelligence Mobile allows you to view Oracle BI EE content on supported mobile devices such as the Apple iPhone and Apple iPad.

Using Oracle Business Intelligence Mobile, you can view and analyze BI content such as analyses and dashboards, BI Publisher content, scorecard content, and content delivered by agents.

**Oracle Business Intelligence Web Services**

See [Oracle Business Intelligence Session-Based Web Services](#) and [Oracle Business Intelligence Web Services for SOA](#).

**Oracle Business Intelligence Session-Based Web Services**

An API that implements SOAP. These web services are designed for programmatic use, where a developer uses one web service to invoke many different business intelligence objects. These web services provide functionality on a wide range of Presentation Services operations. These web services enable the developer to extract results from Oracle BI Presentation Services and deliver them to external applications, perform Presentation Services management functions, and execute Oracle Business Intelligence alerts (known as Intelligent Agents).

See also [Oracle Business Intelligence Web Services for SOA](#).

**Oracle Business Intelligence Web Services for SOA**

Contains three web services, `ExecuteAgent`, `ExecuteAnalysis`, and `ExecuteCondition`, which are hosted by the bimiddleware J2EE application. These web services are designed to enable developers to use third-party web services clients (for example, Oracle SOA Suite) to browse for and include business intelligence objects in service oriented architecture components.

See also [Oracle Business Intelligence Session-Based Web Services](#).

**Oracle Call Interface (OCI)**

A connection interface that the Oracle BI Server can use to connect to Oracle Database data sources. You should always use OCI when importing metadata from or connecting to an Oracle Database.

**Oracle OLAP**

Oracle Database has an OLAP Option that provides an embedded, full-featured online analytical processing server.
Oracle Business Intelligence supports Oracle OLAP as a data source. When you import metadata from an Oracle OLAP source, the Oracle OLAP objects appear in the Physical layer of the Administration Tool. Oracle OLAP objects include Analytic Workspaces, which are containers for storing related cubes.

**Oracle Process Manager and Notification Server (OPMN)**

A process management tool that manages all system components (server processes), and supports both local and distributed process management, automatic process recycling and the communication of process state (up, down, starting, stopping). OPMN detects process unavailability and automatically restarts processes.

See also **system components**.

**Oracle Scorecard and Strategy Management**

A performance management tool that lets you describe and communicate your business strategy. You can drive and assess your corporate strategy and performance from the top of your organization down, or from the bottom up.

**Oracle Technology Network (OTN)**

A repository of technical information about Oracle's products where you can search for articles, participate in discussions, ask the user community technical questions, and search for and download Oracle products and documentation.

**outer edge**

The outer edges are the parts of a trellis view that border the inner graphs. These include the column and row headers, the section headers, and so on.

See also **trellis** and **inner graph**.

**parent-child hierarchy**

A hierarchy of members that all have the same type. All the dimension members of a parent-child hierarchy occur in a single data source. In a parent-child hierarchy, the inter-member relationships are parent-child relationships between dimension members.

See also **dimension**.

**parent-child relationship table**

A table with values that explicitly define the inter-member relationships in a parent-child hierarchy. Also called a closure table.

**pass-through calculation**

A calculation that is not computed by the Oracle BI Server but instead is passed to another data source. Enables advanced users to leverage data source features and functions without the need to modify the Oracle BI repository.

**performance tile**

A view type that displays a single aggregate measure value in a manner that is both visually simple and prominent, yet it immediately reveals summary metrics to the user that will likely be presented in more detail within a dashboard view. Performance tile views communicate status through simple formatting by using color, labels, and limited styles, or through conditional formatting of the background color or measure value to make the tile visually prominent.
permissions
Specify which users can access an object, and limit how users can interact with an object. Examples of permissions include write, delete, and change permissions. See object properties.

perspective
A category in your organization with which to associate initiatives, objectives, and KPIs in a scorecard. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).
See also initiative, key performance indicator (KPI), objective, and Oracle Scorecard and Strategy Management.

physical catalog
An object in the Physical layer of an Oracle BI repository that groups different schemas. A catalog contains all the schemas (metadata) for a database object.

physical display folder
Folders that organize objects in the Physical layer of an Oracle BI repository. They have no metadata meaning.

physical join
Joins between tables in the Physical layer of an Oracle BI repository.

Physical layer
A layer of the Oracle BI repository that contains objects that represent physical data constructs from back-end data sources. The Physical layer defines the objects and relationships available for writing physical queries. This layer encapsulates source dependencies to enable portability and federation.

physical schema
An object in the Physical layer of an Oracle BI repository that represents a schema from a back-end database.

physical table
An object in the Physical layer of an Oracle BI repository, usually corresponding to a table that exists in a physical database.
See also Physical layer.

point of view area
An area within KPIs and scorecards that shows data of specific interest to you, such as the area of business for which you are responsible. The point of view area displays controls for the dimensions of KPIs that are used in the scorecard to measure the progress and performance of initiatives and objectives.
See also initiative, key performance indicator (KPI), and objective.

presentation hierarchy
An object in the Presentation layer of an Oracle BI repository that provides an explicit way to expose the multidimensional model in Presentation Services and other clients. Presentation hierarchies expose analytic functionality such as member selection,
custom member groups, and asymmetric queries. Users can create hierarchy-based queries using presentation hierarchies.

In Presentation Services, presentation hierarchies are displayed as hierarchical columns.

See also hierarchical column and presentation level.

**Presentation layer**
Provides a way to present customized, secure, role-based views of a business model to users. It adds a level of abstraction over the Business Model and Mapping layer in the Oracle BI repository. The Presentation layer provides the view of the data seen by users who build analyses in Presentation Services and other client tools and applications.

See also Business Model and Mapping layer.

**presentation level**
In the Oracle BI repository, a component of a presentation hierarchy that either rolls up or is rolled up from other levels. Presentation levels are displayed as levels within hierarchical columns in Presentation Services.

See also hierarchy level and presentation hierarchy.

**Presentation Services**
See Oracle BI Presentation Services.

**Presentation Services server**
See Oracle BI Presentation Services server.

**presentation table**
An object in the Presentation layer of an Oracle BI repository that is used to organize columns into categories that make sense to the user community. A presentation table can contain columns from one or more logical tables. The names and object properties of the presentation tables are independent of the logical table properties.

**primary key**
A column (or set of columns) where each value is unique and identifies a single row of a table.

**process instance**
A unique process on an individual workstation that is associated with a BI instance.

See also BI instance.

**prompt**
A type of filter that enables the content designer to build and specify data values or the end user to choose specific data values to provide a result sets for an individual analysis or multiple analyses included on a dashboard or dashboard page. A prompt expands or refines existing dashboard and analysis filters.

The types of prompts are column prompts, currency prompts, image prompts, and variable prompts.

See also column prompt, currency prompt, filter, image prompt, and variable prompt.
prompted link
Captures the path to a dashboard page and a simplified presentation of the dashboard prompt.
See bookmark link.

query
Contains the underlying SQL statements that are issued to the Oracle BI Server. You do not have to know a query language to use Oracle Business Intelligence.

query cache
A facility to store query results for use by other queries.

ragged hierarchy
See unbalanced hierarchy.

report
The response returned to the user from the execution of a query created using Oracle BI Publisher. Reports can be formatted, presented on a dashboard page, saved in the Oracle BI Presentation Catalog, and shared with other users.
See also analysis.

repository
See Oracle BI repository.

repository variable
See variable.

results
The output returned from the Oracle BI Server for an analysis.
See also analysis.

scorecard
See Oracle Scorecard and Strategy Management.

selection step
A choice of values that is applied after the query is aggregated that affects only the members displayed, not the resulting aggregate values. Along with filters, selection steps restrict the results for an analysis.
See also analysis and filter.

session variable
See variable.

simple trellis
A trellis view that displays inner visualizations that are all the same type, such as all scatter graphs. The inner visualizations all use a common axis, also known as a synchronized scale.
See also trellis, synchronized scale, and visualization.
skip-level hierarchy
A hierarchy where some members do not have a value for a particular ancestor level. For example, in the United States, the city of Washington in the District of Columbia does not belong to a state. The expectation is that users can still navigate from the country level (United States) to Washington and below without the need for a state.
See also hierarchy.

smart watchlist
A smart watchlist is a view into a particular scorecard based on criteria that you specify. For example, a smart watchlist might show the top ten KPIs in a scorecard based on best performance or all the objectives, initiatives, and KPIs in a scorecard that are owned by a specific business owner.
See also watchlist and Oracle Scorecard and Strategy Management.

snowflake schema
A dimensional schema where one or more of the dimensions are partially or completely normalized.

spark graph
An embedded mini-graph that, in conjunction with other mini-graphs and numbers, illustrates a single trend. Spark graphs are also known as sparks.
Sparks do not include axes or labels; they get their context from the content that surrounds them. Each type of spark graph has only one measure, which is hidden; the scale is relative to itself only.
A spark graph can be of the graph subtype Spark Line, Spark Bar, or Spark Area.
See also microchart.

SQL
See structured query language (SQL).

star schema
A relational schema that allows dimensional analysis of historical information. Star schemas have one-to-many relationships between the logical dimension tables and the logical fact table. Each star consists of a single fact table joined to a set of denormalized dimension tables.

strategy contribution wheel
A component of a scorecard that makes it easy to see the contribution (or impact) a specific objective or KPI has on a parent objective in a series of concentric rings (or wheel diagram). You use the strategy contribution wheel diagram to hierarchically view an objective and its supporting child objectives and KPIs.
See also Oracle Scorecard and Strategy Management.

strategy map
A component of a scorecard that shows how the objectives that have been defined for a scorecard and the KPIs that measure their progress are aligned by perspectives. It also shows cause and effect relationships.
See also Oracle Scorecard and Strategy Management.
**strategy tree**
A component of a scorecard that shows an objective and its supporting child objectives and KPIs hierarchically in a tree diagram.
See also Oracle Scorecard and Strategy Management.

**structured query language (SQL)**
A standard programming language for querying and modifying data.
See also Logical SQL.

**subject area**
In an Oracle BI repository, an object in the Presentation layer that organizes and presents data about a business model. It is the highest-level object in the Presentation layer and represents the view of the data that users see in Presentation Services. Oracle BI repository subject areas contain presentation tables, presentation columns, and presentation hierarchies.
In Presentation Services, subject areas contain folders, measure columns, attribute columns, hierarchical columns, and levels.

**synchronized scale**
(Applicable to simple trellis only) A synchronized scale means that all the visualizations within the trellis are viewed on the same scale, that is, they share a common axis. Having a common axis makes all graph markers easy to compare across rows and columns.
See also simple trellis and visualization.

**system components**
Server processes (not Java applications) that are managed by the Oracle Process Manager and Notification server (OPMN).
See also Oracle Process Manager and Notification Server (OPMN).

**transformation**
Work that is performed on data when moving from a database to another location (sometimes another database). Some transformations are typically performed on data when it is moved from a transaction system to a data warehouse system.

**trellis**
Displays multidimensional data shown as a set of cells in a grid, where each cell represents a subset of data using a particular graph type. Data can be represented with graphs, microcharts, and numbers.
The trellis view has two subtypes: simple trellis and advanced trellis.
See also advanced trellis and simple trellis.

**unbalanced hierarchy**
A hierarchy where the leaves do not have the same depth. For example, an organization might choose to have data for the current month at the day level, data for the previous year at the month level, and data for the previous five years at the quarter level.
See also hierarchy.
value hierarchy
See parent-child hierarchy.

variable
Objects in an Oracle BI repository that are used to streamline administrative tasks and dynamically modify metadata content to adjust to a changing data environment.

Variables are of the following types:

- Repository variables have a single value at any point in time. There are two types of repository variables: static and dynamic.
- Session variables are created and assigned a value when each user logs on. There are two types of session variables: system and nonsystem.

variable prompt
Enables the user to select a value specified in the variable prompt to display on the dashboard. A variable prompt is not dependent upon column data, but enables you to manipulate, for example add or multiply, the column data on an analysis.

See also prompt.

virtual physical table
A physical table that is made from a stored procedure or a \texttt{SELECT} statement. Creating virtual tables can provide the Oracle BI Server and the underlying databases with the proper metadata to perform some advanced query requests.

vision statement
A short statement in a scorecard that describes what your organization wants to become sometime in the future. For example, it might be to become the most successful business in the South America Polypropylene Market.

See also mission statement and Oracle Scorecard and Strategy Management.

visualization
In the context of Oracle BI EE, a visualization is the choice of graph that appears within a data cell in a trellis view. There are many visualizations from which to choose when creating a trellis view, including bar graphs, scatter graphs, and spark graphs.

See also trellis.

watchlist
A table that lists scorecard objects (that is, initiatives, objectives, and KPIs) that are related to a particular aspect of a scorecard or that are grouped together for a particular purpose. There are different types for watchlists for example, KPI watchlists or smart watchlists.

See also KPI watchlist, smart watchlist, and Oracle Scorecard and Strategy Management.

waterfall graph
A graph type that lets you visualize how a value increases or decreases sequentially and cumulatively. Waterfall graphs focus the user’s attention on how each measure contributes to the overall total and communicate through simple formatting by using color.
WebLogic server domain
Contains Java components that are configured to participate in the servicing of SOAP, HTTP, and other forms of requests.

WebLogic Scripting Tool (WLST)
A command-line scripting interface that enables you to configure, manage, and persist changes to WebLogic Server instances and domains and to monitor and manage server runtime events.

XML API
See Oracle BI Server XML API.
Index

A
Administration Tool, using for repository development, 1-11
aggregation rule element
  in MDS XML schema, 2-21
  in XML schema, 7-18
APIs
  Oracle BI Server XML API, 6-1
appending objects, using Oracle BI Server XML API, 6-2
application roles
  managing, 1-11

B
  bi-init, 1-3, 1-5, 6-2, 6-3
  biserverxmlcli command-line tool, using, 6-5
  biserverxmlmexec command-line tool, using, 6-4
  biserverxmlgen command-line tool
    for generating MDS XML, 1-4
    for generating XML, 6-3
business model element
  in MDS XML schema, 2-3
  in XML schema, 7-3

C
catalog link element
  in MDS XML schema, 5-1
  in XML schema, 10-1
certification information, 1-11
column mapping element
  in MDS XML schema, 2-18
  in XML schema, 7-16
command-line tools
  biserverxmlcli, 6-5
  biserverxmlmexec, 6-4
  biserverxmlgen, 1-4, 6-3
complex join element
  in MDS XML schema, 3-20
  in XML schema, 8-18
connection pool element
  in MDS XML schema, 3-12
  in XML schema, 8-11
connection pool settings, updating using Oracle BI Server XML API, 6-10
cube column element
  in MDS XML schema, 3-29
  in XML schema, 8-27
cube table element
  in MDS XML schema, 3-21
  in XML schema, 8-19
custom authenticator element
  in MDS XML schema, 5-14
  in XML schema, 10-16

D
data sources
  supported by Oracle Business Intelligence, 1-11
database element
  in MDS XML schema, 3-1
  in XML schema, 8-1
databases, supported, 1-11
DB sign-on element
  in MDS XML schema, 5-24
  in XML schema, 10-24
dimension element
  in MDS XML schema, 2-7
  in XML schema, 7-6
DSNs, setting up for the Oracle BI Server, 1-11

generic elements and attributes, for Oracle BI Server XML API, 1-6, 6-13
group element
  in MDS XML schema, 5-23
  in XML schema, 10-23
groups, managing, 1-11

H
hardware requirements, 1-11

I
id, about, 6-15
init block element
  in MDS XML schema, 5-11
  in XML schema, 10-13
installation types, supported, 1-11

J
JDKs, supported, 1-11

L
LDAP server element
  in MDS XML schema, 5-9
  in XML schema, 10-11
list catalog element
  in MDS XML schema, 5-4
  in XML schema, 10-5
logging, managing, 1-11
logical column element
  in MDS XML schema, 2-1
  in XML schema, 7-1
logical complex join element
  in MDS XML schema, 2-4
  in XML schema, 7-4
logical display folder element
  in MDS XML schema, 2-23
  in XML schema, 7-20
logical elements, for Oracle BI Server MDS XML API, 2-1
logical elements, for Oracle BI Server XML API, 7-1
logical foreign key element
  in MDS XML schema, 2-12
  in XML schema, 7-9
logical key element
  in MDS XML schema, 2-13
  in XML schema, 7-11
logical level element
  in MDS XML schema, 2-5
  in XML schema, 7-5
logical source folder element
  in MDS XML schema, 2-19
  in XML schema, 7-17
logical table element
  in MDS XML schema, 2-9
  in XML schema, 7-8
logical table source element
  in MDS XML schema, 2-14
  in XML schema, 7-12

M
marketing elements, for Oracle BI Server MDS XML API, 5-1
marketing elements, for Oracle BI Server XML API, 10-1
MDS XML APIs
  Oracle BI Server MDS XML API, 1-1
MDS XML schema
  aggregation rule element, 2-21
  business model element, 2-3
catalog link element, 5-1
column mapping element, 2-18
complex join element, 3-20
cube column element, 3-29
cube table element, 3-21
custom authenticator element, 5-14
database element, 3-1
DB sign-on element, 5-24
dimension element, 2-7
group element, 5-23
init block element, 5-11
LDAP server element, 5-9
list catalog element, 5-4
logical column element, 2-1
logical complex join element, 2-4
logical display folder element, 2-23
logical foreign key element, 2-12
logical key element, 2-13
logical level element, 2-5
logical source folder element, 2-19
logical table element, 2-9
logical table source element, 2-22
measure definition element, 2-20
object privilege element, 5-18
override aggregation rule element, 2-22
physical catalog element, 3-4
physical column element, 3-11
physical dimension element, 3-24
physical display folder element, 3-5
physical foreign key element, 3-19
physical hierarchy element, 3-26
physical key element, 3-18
physical level element, 3-28
physical schema element, 3-4
physical table element, 3-6
presentation catalog element, 4-1
presentation column element, 4-5
presentation hierarchy element, 4-7
presentation level element, 4-9
presentation table element, 4-3
privilege package element, 5-15
project element, 5-24
qualified list item element, 5-4
qualifying key element, 5-6
query object element, 5-28
query privilege element, 5-20
sampling table element, 5-6
seg catalog element, 5-7
target level element, 5-2
target privilege package element, 5-21
variable element, 5-8
MDS XML, generating and executing with Oracle BI Server MDS XML API, 1-2
mdsid, about, 1-7
measure definition element
  in MDS XML schema, 2-20
  in XML schema, 7-17
memory requirements, 1-11
migrating from test to production environments, 6-10
migrating metadata, with Oracle BI Server XML API, 6-6
minimum disk space requirements, 1-11
N
naming restrictions for repository objects, 6-15
NQSConfig.INI
using, 1-11

O
object identifiers, for Oracle BI Server MDS XML
API, 1-7
object identifiers, for Oracle BI Server XML
API, 6-14
object privilege element
in MDS XML schema, 5-18
in XML schema, 10-19
ODBC DSN
for the Oracle BI Server, 1-11
operating systems, supported, 1-11
Oracle BI repository
maintaining versions with Oracle BI Server XML
API, 6-6
programmatically updating connection pool
settings, 6-10
programmatically updating using Oracle BI Server
XML API, 6-7
Oracle BI repository, designing and building, 1-11
Oracle BI Server
ODBC DSN configuration, 1-11
starting and stopping, 1-11
Oracle BI Server MDS XML API
about, 1-1
about object identifiers, 1-7
command-line tool, 1-3
Oracle BI Server MDS XML API elements
 logical elements, 2-1
 marketing elements, 5-1
 other elements, 5-24
 physical elements, 3-1
 presentation elements, 4-1
 security elements, 5-8
Oracle BI Server Web services, using, 1-11
Oracle BI Server XML API
about, 6-1
about object identifiers, 6-14
applying patches to online repositories, 6-7
command-line tools, 6-2
maintaining RPD versions with, 6-6
programmatically updating repositories, 6-7
using to append objects, 6-2
using to migrate metadata, 6-6
using to migrate to production
 environments, 6-10
using XSL with, 6-12
Oracle BI Server XML API elements
generic elements and attributes, 1-6, 6-13
 logical elements, 7-1
 marketing elements, 10-1
 other elements, 10-27
 physical elements, 8-1
 presentation elements, 9-1
 security elements, 10-10
Oracle Business Intelligence
logging, 1-11
starting and stopping, 1-11
Oracle WebLogic Server, managing, 1-11
other elements, for Oracle BI Server MDS XML
API, 5-24
other elements, for Oracle BI Server XML API, 10-27
override aggregation rule element
in MDS XML schema, 2-22
in XML schema, 7-19

P
patches, applying to online repositories using Oracle
BI Server XML API, 6-7
permissions
setting using command-line tools, 6-13
physical catalog element
in MDS XML schema, 3-4
in XML schema, 8-4
physical column element
in MDS XML schema, 3-11
in XML schema, 8-9
physical dimension element
in MDS XML schema, 3-24
in XML schema, 8-22
physical display folder element
in MDS XML schema, 3-5
in XML schema, 8-5
physical elements, for Oracle BI Server MDS XML
API, 3-1
physical elements, for Oracle BI Server XML
API, 8-1
physical foreign key element
in MDS XML schema, 3-19
in XML schema, 8-17
physical hierarchy element
in MDS XML schema, 3-26
in XML schema, 8-24
physical key element
in MDS XML schema, 3-18
in XML schema, 8-16
physical level element
in MDS XML schema, 3-28
in XML schema, 8-26
physical schema element
in MDS XML schema, 3-4
in XML schema, 8-4
physical table element
in MDS XML schema, 3-6
in XML schema, 8-5
platforms, supported, 1-11
presentation catalog element
in MDS XML schema, 4-1
in XML schema, 9-1
presentation column element
in MDS XML schema, 4-5
in XML schema, 9-5
presentation elements, for Oracle BI Server MDS XML
API, 4-1
presentation elements, for Oracle BI Server XML API, 9-1
presentation hierarchy element
in MDS XML schema, 4-7
in XML schema, 9-6
presentation level element
in MDS XML schema, 4-9
in XML schema, 9-8
presentation table element
in MDS XML schema, 4-3
in XML schema, 9-3
privilege package element
in MDS XML schema, 5-15
in XML schema, 10-17
processes, starting and stopping, 1-11
production environments, moving to, 1-11
project element
in MDS XML schema, 5-24
in XML schema, 10-27

Q
qualified list item element
in MDS XML schema, 5-4
in XML schema, 10-6
qualifiedName, about, 6-14
qualifying key element
in MDS XML schema, 5-6
in XML schema, 10-8
query object element
in MDS XML schema, 5-28
in XML schema, 10-30
query privilege element
in MDS XML schema, 5-20
in XML schema, 10-20

S
sampling table element
in MDS XML schema, 5-6
in XML schema, 10-8
schema document, for Oracle BI Server XML API, 6-1
security
managing, 1-11
security elements, for Oracle BI Server MDS XML API, 5-8
security elements, for Oracle BI Server XML API, 10-10
seg catalog element
in MDS XML schema, 5-7
in XML schema, 10-9
SetPermission element
in XML schema, 10-25
software requirements, 1-11
source control management, using for repository development, 1-11
starting Oracle Business Intelligence processes, 1-11
stopping Oracle Business Intelligence processes, 1-11
system requirements, 1-11

T
target level element
in MDS XML schema, 5-2
in XML schema, 10-2
test to production, 1-11
third-party products, supported, 1-11

U
uid, about, 6-15
user element
in MDS XML schema, 5-21
in XML schema, 10-22
users, managing, 1-11

V
variable element
in MDS XML schema, 5-8
in XML schema, 10-10

W
Web services, for Oracle BI Server, 1-11

X
XML API
See Oracle BI Server XML API
XML schema
aggregation rule element, 7-18
business model element, 7-3
catalog link element, 10-1
column mapping element, 7-16
common elements and attributes, 1-6, 6-13
complex join element, 8-18
connection pool element, 8-11
cube column element, 8-27
cube table element, 8-19
custom authenticator element, 10-16
database element, 8-1
DB sign-on element, 10-24
dimension element, 7-6
group element, 10-23
init block element, 10-13
LDAP server element, 10-11
list catalog element, 10-5
logical column element, 7-1
logical complex join element, 7-4
logical display folder element, 7-20
logical foreign key element, 7-9
logical key element, 7-11
logical level element, 7-5
logical source folder element, 7-17
logical table element, 7-8
logical table source element, 7-12
measure definition element, 7-17
object privilege element, 10-19
override aggregation rule element, 7-19
physical catalog element, 8-4
physical column element, 8-9
physical dimension element, 8-22
physical display folder element, 8-5
physical foreign key element, 8-17
physical hierarchy element, 8-24
physical key element, 8-16
physical level element, 8-26
physical schema element, 8-4
physical table element, 8-5
presentation catalog element, 9-1
presentation column element, 9-5
presentation hierarchy element, 9-6
presentation level element, 9-8
presentation table element, 9-3
privilege package element, 10-17
project element, 10-27
qualified list item element, 10-6
qualifying key element, 10-8
query object element, 10-30
query privilege element, 10-20
sampling table element, 10-8
seg catalog element, 10-9
SetPermission element, 10-25
target level element, 10-2
user element, 10-22
variable element, 10-10
XML, generating and executing with Oracle BI Server
XML API, 6-2
XSL, using with Oracle BI Server XML API, 6-12
xudml_mds_admin.xsd file, 1-1
xudml_mds_core.xsd file, 1-1
xudml1.xsd file, 6-1