Oracle® Fusion Middleware

Developer's Guide for Oracle Business Intelligence Enterprise Edition

11*g* Release 1 (11.1.1)

E10545-01

July 2010

Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Enterprise Edition, 11*g*
Release 1 (11.1.1)

E10545-01

Contents

2 Using the Oracle BI EE Logical SQL View Object

3 Integrating Oracle Business Intelligence Presentation Services With Oracle WebCenter

Index

Preface

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, real-time predictive intelligence, and an enterprise reporting engine. Oracle BI EE is designed to bring greater business visibility and insight to a wide variety of users.

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 is part of the documentation set for Oracle BI EE. This guide contains information about developing Oracle BI EE-based applications and integrations between Oracle BI EE and other Oracle systems.

Audience

This document is intended for application developers, data service providers, and middle tier administrators who want to programatically access and use the Oracle BI EE components to create applications or integrations with other components.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/support/contact.html` or visit `http://www.oracle.com/accessibility/support.html` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Business Intelligence Enterprise Edition 11*g* Release 1 (11.1.1) documentation set:

- *Oracle Fusion Middleware Release Notes* for your platform
- *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence* for your platform
- *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*
- *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

New Features for Oracle Business Intelligence Developers

This section describes new features in Oracle Business Intelligence 11*g* Release 1 (11.1.1).

This preface contains the following topics:

- "New Features"
- "Upgrade Considerations"
- "System Requirements and Certification"

New Features

Embedding Business Intelligence Objects Into ADF Applications

You can add business intelligence objects stored in the Oracle BI EE Presentation Services Catalog to ADF applications. This functionality is delivered through Oracle BI EE extensions that you plug into Oracle JDeveloper. Any filters, prompts, or action links included in business intelligence objects will work as designed when included in an ADF application.

For information, see Chapter 1, "Embedding Business Intelligence Objects in ADF Applications".

Oracle BI EE Logical SQL View Object

When creating ADF applications, you can use the Business Intelligence Logical SQL View Object to access the Oracle Business Intelligence Server, fetch Oracle BI EE data, and bind that data to native ADF components. This method requires you to create a JDBC connection to the Oracle Business Intelligence Server. The functionality to build the BI JDBC connection is standard JDeveloper functionality.

The remaining functionality required to bind the data to ADF view objects is delivered through Oracle BI EE extensions that you plug into JDeveloper.

For information, see Chapter 2, "Using the Oracle BI EE Logical SQL View Object".

Integrating the Oracle BI EE Presentation Services Catalog With WebCenter Applications

This release of Oracle BI EE supports integration with Oracle WebCenter. You can use JDeveloper to create and configure a WebCenter application that integrates with the Oracle Business Intelligence Presentation Services Catalog so that at runtime, users can open and browse the catalog and add business intelligence objects to their WebCenter Spaces pages.

For information, see Chapter 3, "Integrating Oracle Business Intelligence Presentation Services With Oracle WebCenter".

Upgrade Considerations

Because this guide covers only new features and functionality, there are no items that you need to be aware of if your site has upgraded from Oracle Business Intelligence 10*g* to Oracle Business Intelligence 11*g* Release 1 (11.1.1.1).

For more information about upgrading to Oracle Business Intelligence 11*g*, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*.

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:

http://www.oracle.com/technology/software/products/ias/files/fusion_requirements.htm

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

http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html

1 Embedding Business Intelligence Objects in ADF Applications

This chapter explains how to add business intelligence objects to ADF applications. Examples of business intelligence objects include analyses, dashboards, and Scorecard components.

This chapter includes the following sections:

- "Overview of Embedding Business Intelligence Objects in Applications"
- "Business Intelligence Objects That You Can Add to ADF Pages"
- "Embedded Business Intelligence Objects Use Cases"
- "Preparing the ADF Project to Accept Business Intelligence Objects"
- "Creating an Oracle BI EE Presentation Services Connection"
- "Adding Oracle Business Intelligence Objects to ADF Pages"
- "Passing Business Intelligence Content with the Oracle BI EE Contextual Event Action"
- "Adding or Modifying a Presentation Services Connection After Deployment"
- "Credentials for Connecting to the Oracle BI EE Presentation Catalog"

1.1 Overview of Embedding Business Intelligence Objects in Applications

Embedding business intelligence objects allows you to access the Oracle BI EE Presentation Services catalog and choose Oracle BI EE objects, such as analyses and dashboards, to include in ADF pages. In most cases, the objects that you add to ADF pages are created to fit the theme of an ADF page and application. When you add a business intelligence object to an ADF page, that page will contain a reference to the business intelligence object and does not contain a copy of the object. When the business intelligence object is modified and saved to the Oracle BI EE Presentation Services Catalog, any changes will appear in the ADF application when the user runs the form.

After the business intelligence object is added to the ADF page, you can wire it to other regions on the ADF page. You can also set any filter or prompt parameters for the object. You can also set up security and change the Presentation Services connection after the application is deployed.

Note that the Business Intelligence Logical SQL View Object allows you to access the Oracle Business Intelligence Server to create ADF applications that bind native ADF components to Oracle BI EE data. For more information about this method of adding business intelligence data to ADF pages, see Chapter 2, "Using the Oracle BI EE Logical SQL View Object".

1.2 Business Intelligence Objects That You Can Add to ADF Pages

The business intelligence objects that you can include in ADF applications are stored in the Presentation Services Catalog's folders. You can include the following types of business intelligence objects in ADF applications:

- Analyses (Note that you can expand and browse the folders to view an analysis' subcomponents. The following view types appear in the catalog: table, pivot table, chart, funnel chart, gauge, narrative, ticker and title.)
- Dashboards (Note that you can browse inside the dashboard folder to find the pages that are associated with the dashboard.)
- Dashboard Pages (Note that you cannot browse within the dashboard pages to see components. Such as, any analyses embedded in the dashboard.)
- Scorecard components (Note that you cannot include an entire Scorecard; only its components. These components are: strategy maps, strategy trees, KPI watchlists, cause and effect maps, and custom views.)

1.3 Embedded Business Intelligence Objects Use Cases

Use the following use cases to help you better understand why you would embed business intelligence objects in ADF applications. For specific information about creating the business intelligence objects and adding them to an ADF application, see the procedural topics in this chapter as well as *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* and *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

- Supply chain managers want to view the analysis summarizing the status of orders in different sectors and availability of items in the warehouse along with other transactional data.
- A marketing manager for an online shopping site wants to see buyers' trend information, such as age group, geographic distribution, and personal interests, in a dashboard to analyze the user data and use it to determine future marketing strategies.
- A sales manager wants to see the sales report for different regions with conditional formatting and take an action if the sales is trending below a minimum threshold. This report with an action link can be created in the Oracle BI EE application and embedded into the ADF application.

1.4 Preparing the ADF Project to Accept Business Intelligence Objects

To properly prepare the ADF project for the addition of business intelligence objects, you must add the required MDS libraries (MDS Runtime and MDS Runtime Dependencies) to the ADF project and add the BI ADF Components technology scope to the ADF project.

1.4.1 How to Prepare the ADF Project

Use the following procedures to prepare the ADF project to accept business intelligence content. Before you can perform the following tasks, you must have created an ADF project to which you will add business intelligence content.

When you have completed the procedures in this topic, you can create an Oracle BI Enterprise Edition Presentation Services connection to access the catalog from which to select business intelligence objects to embed in your application. For more information, see "Creating an Oracle BI EE Presentation Services Connection".

1.4.1.1 Checking for Installed JDeveloper Extensions

Use this procedure if you are not sure whether you have the required Oracle JDeveloper 11g Release 1 (11.1.1) extensions and Oracle BI EE extensions.

1. In JDeveloper, click the **Help** menu and click **About**. The "About Oracle JDeveloper 11g" dialog displays.
2. Click the **Extensions** tab to view a list of the installed extensions. To sort the list, click either the **Name** or **Identifier** column heading.

1.4.1.2 Installing the Required JDeveloper Extensions

Use this procedure to install the required extensions.

1. From the JDeveloper **Help** menu, select **Check for Updates**.
2. Follow the steps in the wizard to locate, download, and install the WebCenter and Oracle BI EE extensions.

1.4.1.3 About Oracle BI EE Personalizations in ADF Applications

Personalizations for dashboards, dashboard pages, reports, and views are saved automatically if the ADF application is configured for personalizations. Personalizations can be stored at the user level or the group level.

Users can personalize the following items when they exist on a dashboard region on an ADF page:

- Values for dashboard prompts
- Values for presentation variables (These variables are usually set by a prompt.)
- Values for column selectors
- Drills, sorts, and pivot/swap on a view

Users cannot personalize the following business intelligence content types:

- Dashboard pages
- Stand-alone Scorecard components

1.4.1.4 Adding the MDS Libraries and Enabling MDS Customizations

If you want to save site and user application personalizations such as prompt values, then you must add the following libraries to the ADF project and confirm that the MDS customizations have been properly enabled. MDS stores application metadata and also stores and retrieves customizations at various levels within the ADF application.

To add the MDS libraries to the ADF project:

1. In JDeveloper, go to the Project Pane, right click the project to which you want to add the MDS libraries, and select **Project Properties**.
2. Select **Libraries and Classpath**.
3. Click the **Add Library** button, highlight **MDS Runtime** and **MDS Runtime Dependencies**, and click **OK**.
4. From the left pane of the Project Properties dialog, select **ADF View**.
5. Confirm that the following options are selected:
   - Enable User Customizations
   - Across Sessions using MDS
   - Enable Seeded Customizations
6. In the Project Properties screen, click **OK**.

1.4.1.5 Adding the ADF Technology Scopes

Perform this procedure to add the proper technology scopes to the ADF project.

Technology scopes are ADF project attributes that are used to identify the different technologies used for the project. Selecting the BI ADF Components technology scope adds the ADF Faces, Java, JSF, and JSP and Servlets technology scopes to the ADF project.

To add the ADF technology scopes to the ADF project:

1. In Oracle JDeveloper, go to the Projects Pane and right-mouse click on the project to which you want to add the technology scopes and select **Project Properties**.
2. Select **Technology Scope**.
3. In the Available Technologies list, select **BI ADF Components** and click the **Add** button. Oracle JDeveloper adds the required BI technologies to the Selected Technologies list.
4. Click **OK**.

1.5 Creating an Oracle BI EE Presentation Services Connection

Before you can add business intelligence objects to an ADF application, you must configure SSL, export and import a certificate, and create a data connection to Oracle BI EE Presentation Services.

1.5.1 How to Configure SSL and Create an Oracle BI EE Presentation Services Connection

Use the following procedures to configure SSL, export and import the certificate, and create a connection to Oracle BI EE Presentation Services. Before you can perform the following tasks, you must have first created an ADF project and prepared it to accept business intelligence content. For more information about preparing the project, see "Preparing the ADF Project to Accept Business Intelligence Objects".

> **Note:** If you receive a certificate error when using the application, you may need to import the client certificate into your browser.

1.5.1.1 Configuring SSL

To enable secure communication, you must configure Oracle BI EE Presentation Services to communicate over SSL.

For more information about SSL and how to configure it in Oracle Business Intelligence, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

1.5.1.2 Exporting and Importing the Client Certificate

After you configure SSL, you must export the Web application client certificate from the key store and import it into Oracle JDeveloper.

To export and import the client certificate:

1. Identify the location of the server key store. This path is usually `ORACLE_HOME/oraclebi/wlserver_10.3/server/lib/DemoIdentity.jks`
2. Using the server key store location, run the following command:

   ```
   cd %ORACLE_HOME%/oraclebi/wlserver_10.3/server/lib/
   ```

   ```
   %JAVA_HOME%\bin\keytool -export -alias wlscertgencab -file
   my.crt -keystore  DemoTrust.jks -storepass
   DemoTrustKeyStorePassPhrase
   ```

3. Import the client certificate into the Oracle JDeveloper IDE. This step prevents SOAP errors from occurring when you create an Oracle Business Intelligence Presentation Services connection.
   1. Copy the certificate into JDeveloper's JRE. The location will be `jdev\jdk\jre\lib\security` or `oracle\jdk\jre\lib\security`
   2. Run the following command to import the certificate. Note that the default password for the Java default keystore file `$JAVA_HOME/lib/security/cacerts` is **changeit**.

      ```
      keytool -v -import -file my.crt -keystore cacerts
      ```

4. Import the client certificate into JDeveloper's JRE. This step prevents errors from occurring when sending SOAP requests for `https://server:port/analytics/saw.dll?privateWSDL` in SSL mode.
   1. Open JDeveloper and locate the -Djavax.net.ssl.trustStore parameter. For example, -Djavax.net.ssl.trustStore= C:\ADE\jdevmain3\oracle\wlserver_10.3\server\lib\DemoTrust.jks.

      Note that this parameter contains the client trustStore location and stores the key and certificate that the client uses to verify the server's certificate.
   2. Run the following command to import the certificate to this client trustStore:

      ```
      %ORACLE_HOME%\oracle\jdk\bin\keytool -import -alias test
      -file <client_cert_filename> -keystore <keystorefile>
      -keypass <key_password>
      ```

1.5.1.3 Creating a connection

You can add connections to one or more installation of Presentation Services. Depending upon where you created and saved the connection, the connection name will appear either within the Resource Palette's IDE Connections panel or the Application Navigator's Application Resources panel.

When you expand the connection's name, JDeveloper shows the Presentation Service catalog's folders and the objects stored in the folders. From the folders, you can drag and drop business intelligence objects from the resource palette to the ADF page.

This connection only needs to be built once per catalog; however, after the connection is built, its parameters can be edited at design time within JDeveloper.

Note: A Presentation Services connection can be added or modified after the ADF application is deployed. For more information, see "Adding or Modifying a Presentation Services Connection After Deployment".

To create a connection

1. In JDeveloper, choose where you want the connection to be used.

   If you want to create the connection to be available for any JDeveloper application, then navigate to the Resource Palette, click the **New** button and select **New Connections**, and then select **BI Presentation Services**. The Create BI Presentation Services Connection wizard displays.

   Note: JDeveloper allows you to drag and drop a saved connection from the Resource Palette to the Application Resources pane.

   If you want to create the connection to be available for the current application, only, then navigate to the Application Navigator tab and then to the Application Resources pane, right click on the **Connections** folder, and select **New Connections** and then select **BI Presentation Services**. The Create BI Presentation Services Connection wizard displays.

2. Enter a unique, descriptive name that identifies the catalog connection. Note that after you finish creating the connection, the name cannot be changed.

3. Click **Next**.

4. Enter the Oracle Business Intelligence Presentation Services location and make selections for the following items:

   - **Protocol**– Choose *http* or *https* from the drop down list.
   - **BI Presentation Services Host**– Type the location of the server where Oracle Business Intelligence is installed, for example, *abc12345.mycompany.com*.
   - **BI Presentation Services Port**– Enter the port number for the host. This number is usually *9704*.
   - **BI Presentation Services Virtual Directory**– Enter *analytics* in lower case.

5. Click **Next**.

6. Specify the user name and password for the Presentation Services connection.

   Note: When deploying the ADF application, you must change the username and password to Impersonate User credentials. For more information, see Section 1.9, "Credentials for Connecting to the Oracle BI EE Presentation Catalog".

   ---

7. Click **Next**
8. Click **Test Connection** to verify that the connection information is correct. If your test connection fails, Oracle JDeveloper will provide you with error messages that will help you troubleshoot the problem.
9. Click **Finish** to create the connection.

1.6 Adding Oracle Business Intelligence Objects to ADF Pages

When you drag and drop a business intelligence object that contains filters or parameters to the JSPX page, JDeveloper inserts the *<adfbi:content>* element. This tag references a biContent element in the page definition file using the following EL expression:

<adfbi:content id="myBIRegion" value="#{bindings.biContent}"/>

At runtime, this element returns the markup for the selected business intelligence component and defines the parameters that the region can accept as input. There is one parameter for each prompt in the underlying analysis, or one parameter for the underlying filter or prompt for the underlying dashboard. The *biContent* element contains the default values, and you can edit the values of these parameters and replace them with Expression Language (EL) or literals. This allows you to use JDeveloper's Expression Language to wire the parameters to other components on the page.

The *biContent* element also accepts the *setParameters* ADF contextual event. You can map the payload for this event to the parameters exposed by the dashboard region in the pagedef file. For example:

```
<eventMap xmlns="http://xmlns.oracle.com/adfm/event">
   <event name="setParameters">
    <producer region="doInputs">
     <consumer handler="biExecBinding1">
      <parameters>
       <parameter name="Products.Brand" value="${payLoad.input1}"/>
       <parameter name='Periods."Year"' value="${payLoad.input2}"/>
      </parameters>
     </consumer>
    </producer>
   </event>
</eventMap>
```

If the dashboard or dashboard page contains analyses that use presentation variables and there are no visible prompts on the dashboard to set these variables, the dashboard designer must add hidden prompts to set the variables. When the dashboard is included on an ADF page, the hidden prompts are available for context passing.

1.6.1 How to Add Business Intelligence Objects to an ADF Page

All business intelligence objects added to the ADF page inherit styles from the current ADF skin.

Before you perform the following tasks, you must have first created an ADF project, prepared it to accept business intelligence content, and created a server connection. For more information about these tasks, see "Preparing the ADF Project to Accept Business Intelligence Objects" and "Creating an Oracle BI EE Presentation Services Connection".

To add business intelligence objects to an ADF page and specify filter or prompt parameters

1. In JDeveloper, go to the Resource Palette, select the data connection, and browse for the business intelligence content that you want to add to the ADF page.
2. Drag and drop the item onto the page.

> **Note:** If new or revised business intelligence content was saved to the Presentation Services Catalog, the Oracle JDeveloper catalog connection must be refreshed so that all new and revised business intelligence content is available.

3. If the item includes prompts or filters, the Parameters screen appears, listing the analysis or dashboard prompts and filters. To specify an override parameter value, place your cursor in the **Value** field and type the parameter value.
4. To either build an expression or bind the content item to a managed bean, click the **Ellipsis** button. The EL window appears. For more information about building EL expressions or using the EL screen for data binding, see *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.
5. Save the JSPX page.
6. In the Projects area, which is located in the left pane, locate the file name for the page upon which you are working, right-click, and select **Run**.

   Note that if any type of security was set for the application, you will be prompted for a user name and password.

1.7 Passing Business Intelligence Content with the Oracle BI EE Contextual Event Action

The Oracle BI EE contextual event action is delivered with Oracle BI EE and uses the Oracle BI EE Action Framework. At design time, the analysis designer determines which column should contain the Oracle BI EE contextual event action, adds an Invoke a Browser Script action type to the column, and browses for and selects USERSCRIPT.ADFAction. At runtime, the Oracle BI EE contextual event action passes content from the business intelligence object (analysis) to another region on the ADF page or another ADF contextual event on the ADF page.

The Oracle BI EE contextual event action takes no parameters and can be conditionally rendered. When the user accesses the ADF page and clicks a data cell containing the Oracle BI EE contextual event action or if the cell is tied to a condition and that condition's threshold has been met, the system generates an Oracle BI EE contextual event action that has a qualified data reference (QDR) of the cell as its payload.

Suppose that the ADF designer wants to create a business intelligence region on an ADF page that contains functionality that will automatically promote any employee whose three previous employee reviews were rated as "Outstanding." To accomplish this task, the business intelligence content designer would include the Oracle BI EE contextual event action on the analysis' Employee Name column and include a label of "Promote."

1.7.1 What Is the Definition of the QDR?

Below is the QDR's definition:

```
/**
   * String representation of QDR with use of internal Ids
   * <pre>
   *  qdr
   *   : '{"_m":' memberGroups (',"_q":'  qdrQualifier)? '}'
   *  memberGroups
   *   : '[' memberGroup (',' memberGroup)* ']'
   *  qdrQualifier
   *   : '{' ('"_t":[' scalarMember ']') | ('"_p":[' ('true' | 'false') ']') '}'
   *  memberGroup
   *   : '"_g":{' dimensionMembers (',"_l":'  groupQualifier)? '}'
   *  groupQualifier
   *   : '"' ('M' | 'G') '"'
   *  dimensionMembers
   *   : '"' IDENT '":' ('[]' | arrayMembers | hierarchyLevelsMembers)
   *  hierarchyLevelsMembers
   *   : '{' hierarchyLevelMembers (, hierarchyLevelMembers)* '}'
   *  hierarchyLevelMembers
   *   : '"' IDENT '":' ('[]' | arrayMembers)
   *  arrayMembers
   *   : '[' member (',' member)* ']'
   *  member
   *   : scalarMember | arrayMembers | qualifiedMember
   *  scalarMember
   *   : string | number | datetime
   *  datetime
   *   : string
   *  qualifiedMember
   *   : '{"_v":' scalarMember ',"_s":"' ('A' | 'E' | 'T' | 'P') '"}'
   *  IDENT
   *   : ('a' .. 'z' | 'A'..'Z' | '0' .. '9')+
   *
   *  _m: member groups
   *  _q: QDR qualifier; _t or _p
   *  _t: target; column identifier
   *  _p: measure target; true or false
   *  _g: member group
   *  _l: member group qualifier; 'M' or 'G'
   *  _v: member value
   *  _s: member special value; 'A', 'E', 'T' or 'P'
   *
   *  Group Qualifiers:
   *  M - Measure Dimension
   *  G - Selection Group Dimension
   *  For such dimensions, the dimension identifier is irrelevant; we
      typically make it "_"
   *  For measure dimension, members are measure columns
   *  For selection group dimension, members are group Id Strings
   *
   *  Qualified Members: Members that are qualified by a special value
      qualifier such as:
   *  A - All
   *  E - Every
   *  T - Display Total Id
   *  P - Path to catalog object
   *
   *  Dimension identifier for hierarchical columns is of the form:
      "dimensionId:hierarchyId"
   *
   *  <b>
*  Examples:
*  </b>
*
*  Region (id "c0") West, East
*  Year (id "c1") 2000
*  {'_m':[{'_g':{'c0':['West','East']}}, {'_g':{'c1':[2000]}}]}
*
*  Region (id "c0") West, East
*  Year (id "c1") 2000
*  Target  Region
*  {'_m':[{'_g':{'c0':['West','East']}}, {'_g':{'c1':[2000]}}], '_q':
    {'_t':['c0']}}
*
*  Geog dimension (id "d1"), Default hierarchy (id "h1"), State Level
   (id "l1")  CA, WA
*  {'_m':[{'_g':{'d1:h1':{'l1':['CA','WA']}}}]}
*
*  Geog dimension (id "d1"), Default hierarchy (id "h1"), State Level
   (id "l1") and City Level (id "l2") CA, WA, SJC
*  {'_m':[{'_g':{'d1:h1':{'l1':['CA','WA'], 'l2':['SJC']}}}]}
*
*  Region (id "c0") West, East
*  Measures Sales (id "c3"), Profit (id "c4")
*  {'_m':[{'_g':{'c0':['West','East']}}, {'_g':{'_':['c3','c4']}, '_l':'M'}]}
*
*  Region (id "c0") West, East
*  Year (id "c1") 2007, MyFavYears
*  {'_m':[{'_g':{'c0':['West','East']}}, {'_g':{'c1':[2007,{'_v':
    'MyFavYears', '_s':'P'}]}}]}
*
*  Geog dimension (id "d1"), Default hierarchy (id "h1"),
    State Level (id "l1") CA, WA and City Level (id "l2") MyFavCities
*  {'_m':[{'_g':{'d1:h1':{'l1':['CA','WA'], 'l2':[{'_v':'MyFavCities',
    '_s':'P'}]}}}]}
*
*  Region (id "c0") West, East
*  Measures Sales (id "c3")
*  Measure Target
*  {'_m':[{'_g':{'c0':['West','East']}}, {'_g':{'_':['c3']}, '_l':'M'}], '
     _q':{'_p':true}}
*  </pre>
*
* @param propIdExprMap          Map of column expressions key'ed by column Id
*/
```

1.7.2 What Is the Signature of the QDR?

Below is the QDR's signature:

```
public static QDR fromString(String qdrStr) throws BISvsException
```

1.8 Adding or Modifying a Presentation Services Connection After Deployment

Oracle BI EE provides an ADF MBean that allows you to add a new connection to a deployed BI ADF or WebCenter Spaces application. You can also modify a deployed application's existing connection. MBeans are deployed with the application and can be accessed post-deployment using Oracle Enterprise Manager.

If you did not configure SSL at design time, you can configure it after deployment. For more information, see "How to Configure SSL and Create an Oracle BI EE Presentation Services Connection".

1.8.1 How to Add or Modify a Presentation Services Connection

Use the following procedures to add or modify a Presentation Services connection used by a deployed application.

1.8.1.1 Enabling MBeans in Your Application

ServletContextListeners register MBeans. If your application is deployed on an Oracle WebLogic Server that was configured with Java Required Files (JRF), the ADF MBean jars and their dependencies are already part of the system classpath and, therefore, you do not need to package other jars with the application.

Use the following step to enable MBeans in your application.

To enable MBeans in Your Application

1. Open Webapps web.xml.
2. Specify the ServletContextListeners by adding the following entry to the web.xml.

```
<listener>
    <listener-class>oracle.adf.mbean.share.connection.
       ADFConnectionLifeCycleCallBack
    </listener-class>
</listener>
<listener>
    <listener-class>oracle.adf.mbean.share.config.
       ADFConfigLifeCycleCallBack
    </listener-class>
</listener>
<listener>
    <listener-class>oracle.bc4j.mbean.
       BC4JConfigLifeCycleCallBack</listener-class>
</listener>
```

1.8.1.2 Adding or Modifying a Presentation Services Connection

Use the following procedure to configure the connection after the application was deployed.

1. Open Oracle Enterprise Manager and select your WebLogic domain.
2. If your application is deployed in the BI ADF domain, select the **Application Deployments <your domain>**tree node.

   If your application is deployed in the WebCenter Spaces domain, select the **Web Center** tree node, and then select the **WebCenter Spaces** tree node, and then the **webcenter (WLS_Spaces)** tree node.
3. From the list, select **System MBean Browser**. The System MBean Browser pane displays within the Enterprise Manager page.
4. In the System MBean Browser pane, navigate to the ADF Connections tree node by following the below path:
   1. Select the **Application Defined MBeans** tree node.
   2. Select the **oracle.adf.share.connections** tree node.

   3. Select the **Server: <my server name>** tree node. For example, Server:DefaultServer or Server:WLS_Spaces.
   4. Select the **Application:<your application's name>** tree node. For example, Application:Application2 or Application:webcenter.
   5. Select the **ADF Connections** tree node.
   6. Select the **ADF Connections** tree node. The corresponding MBean information displays in the Application Defined MBean pane.

5. In the Application Defined MBean pane, click the **Operations** tab and then click **createConnection/findOrCreateConnection** to create a Presentation Services connection.The Operation:createConnection dialog displays.
6. Specify the required values for the connection. In the connection type **Value** field, type BISoapConnection. Type a name in the connection name **Value** field. Click **Invoke** to create the connection.
7. In the System MBean Browser pane, click **Refresh** to refresh the tree so that the new connection displays.
8. To modify the connection, locate it in the System MBean Browser pane and click it. The connection's information displays in the Application Defined MBean pane.
9. Click the **Attributes** tab and enter the BISoap connection information. Click the **Apply** button to apply your changes.
10. Click the **Operations** tab and click **save** to save the connection.

1.9 Credentials for Connecting to the Oracle BI EE Presentation Catalog

At design time, you need to specify credentials to connect to the Presentation Services Catalog. These credentials are used to retrieve the list of business intelligence objects (for example, analyses, dashboards, and scorecard components) from the Presentation Services Catalog. For example, if you create the connection using the credentials John.Smith/Password, the catalog will present you with the items available to John.Smith.

After you deploy the application, you can set the location of the Presentation Services Catalog to the catalog used in the deployed environment. At deployment time, you can create and add the Impersonate User credentials to the Oracle BI EE Presentation Services connection. After this is set up and a user accesses an ADF page containing business intelligence objects, Oracle BI EE uses the Impersonate User credentials to connect to the Presentation Services Catalog and impersonate the logged-in user.

This process ensures that the login to the Presentation Server is the same as the current user of the application and any access checks are performed as the current user, and data is fetched as the current user. If the ADF page contains business intelligence objects to which the user does not have access, the ADF page returns a message stating that the user does not have the proper permissions to access these objects.

1.9.1 How to Create and Use Impersonate User

Use the following procedures to create and use the BIImpersonateUser user to secure an application that uses an Oracle BI EE Presentation Services connection and includes Oracle BI EE objects.

The Impersonate User feature secures applications that contain Oracle BI EE objects when Oracle BI EE and ADF are not sharing an Oracle Internet Directory (OID). Before you begin the process of creating and using Impersonate User, you must confirm that this capability is configured in your environment.

1.9.1.1 Create the BIImpersonateUser

Before you perform this procedure, make sure that either you or the Administrator have created users in the WebLogic Server's Oracle BI EE realm and assigned the BIConsumer role to each user in this realm.

Use the following procedure to use WebLogic Server Console to create the BIImpersonateUser user.

1. Open WebLogic Server for your Oracle BI EE instance.
2. Locate the Domain Structure pane and select **Security Realm**. The Realms pane displays.
3. In the Realms pane, select **<myrealm>**. The Settings dialog displays.
4. In the Settings dialog, select the **Users and Groups** tab.
5. Confirm that the Users tab is displaying and click **New**.
6. Specify BIImpersonateUser for the user name and enter a password.
7. Click **OK**.

1.9.1.2 Grant Permissions to BIImpersontateUser

User the following procedure to use Oracle Enterprise Manager to grant permissions to BIImpersontateUser.

1. Open Oracle Enterprise Manager for your Oracle BI EE instance.
2. Locate and select bifoundation_domain. The bifoundation_domain pane displays.
3. In the bidoundation_domain pane, click the **WebLogic Domain** list, highlight Security, and select **Application Policies**. The Search pane displays.
4. In the **Select Application Name to Search** field's list, select **AdminServer/obi**.
5. Click **Create**. The Add Permission dialog displays.
6. In the **Permission Class** field, select oracle.security.jps.ResourcePermission. In the **Resource Name** field, select oracle.bi.server.impersonateUser. Click **OK**.
7. In the Grantees pane, click **Add User**. The Add User dialog displays.
8. In the Available Users list, select BIImpersonateUser and click **Move** to move it to the Selected Users list. Click **OK**.
9. In the bifoundation_domain pane, click **OK**.

1.9.1.3 Restart Servers

Use the following procedure if the changes that you made do not display.

1. Stop and restart the following servers:
   - Oracle BI EE Server
   - Oracle BI EE Presentation Server
   - WebLogic Server

2

Using the Oracle BI EE Logical SQL View Object

This chapter explains how you use the Oracle BI EE Logical SQL view object to embed business intelligence data into an ADF application's components.

This chapter includes the following sections:

- "Introduction to the Oracle BI EE Logical SQL View Object"
- "Oracle Business Intelligence Logical SQL View Object Use Cases"
- "Creating an Oracle Business Intelligence Logical SQL View Object"

2.1 Introduction to the Oracle BI EE Logical SQL View Object

The Business Intelligence Logical SQL View Object allows you to access the Oracle Business Intelligence Server to create ADF applications that bind native ADF components to Oracle BI EE data.

The Oracle Business Intelligence view object exposes the items in the select clause as attributes. After the Business Intelligence Logical View Object's setup is complete, you can use the Oracle Business Intelligence Logical SQL View Object wherever the other view objects can be used in an ADF application. For example, you could create a view link, which allows the you to join the attributes of the Oracle Business Intelligence view object to attributes of another view object that returns data from the operational schemas. You can easily show the data from the Oracle Business Intelligence view object, or joined data created using view links, in any ADF rich/faces component by using a data control that JDeveloper automatically generates.

Including business intelligence objects such as analyses, dashboards,and Scorecard components in ADF pages allows you to access the Oracle BI EE Presentation Services catalog, browse the catalog, and choose Oracle BI EE objects to include in ADF pages. For more information about this method of adding business intelligence data to ADF pages, see Chapter 1, "Embedding Business Intelligence Objects in ADF Applications".

2.2 Oracle Business Intelligence Logical SQL View Object Use Cases

The following use cases are designed to help you understand how you can use the Oracle Business Intelligence Logical SQL View Object to create robust ADF pages. Each use case consists of a scenario and general instructions.

2.2.1 Displaying Data from Logical SQL in an ADF-Rich User Interface Component

This use case describes how to display data returned by a logical SQL statement in an ADF face or rich user interface component. This use case assumes that you are interested in displaying data for Revenue by Region and District in a read-only rich table format.

> **Tip:** This use case describes one way of displaying data in a rich table. Normally, it is easier to embed an analysis into an ADF page.

To display data from logical SQL in an ADF-rich user interface component:

1. Create a JDBC connection to an Oracle BI EE Server that contains the data you want to display in the ADF page. For more information, see "Creating a BI JDBC Connection".
2. Create an application module with a standard JDBC connection. For more information, see "Initializing the ADF Project".
3. Create an Oracle Business Intelligence view object, pointing to the JDBC connection and using the following logical SQL. Note that Region, District, and Dollars will be added automatically as the attributes of the View Object.

   ```
   "Select Markets.Region, Markets.District, Sales
   Measures.Dollars from Paint"
   ```

4. Navigate to the Application Model page of the Create View Object wizard and add an instance of the view object that you created to an application module. In this use case, assume that the application module is named "BIVO_AppModule."

   JDeveloper automatically creates a BIVO_AppModuleDataControl. Under this data control your view object is listed as a collection with all its attributes.

5. Drag and drop the view object collection from the data control to the appropriate place on a JSPX page. When the pop-up menu appears, select an appropriate ADF Rich UI component to use for displaying the view object.
6. Select a read-only Rich table for the display.

2.2.2 Displaying Data from Logical SQL and Passing in a Runtime Parameter

This use case is an extension of the "Displaying Data from Logical SQL in an ADF-Rich User Interface Component" use case. In this use case, however, you want to display data for Revenue by Region and District from logical SQL, but only when Revenue is greater than a number inputted by the user. You can accomplish this by passing in a run-time parameter.

To display data from logical SQL and passing in a runtime parameter:

1. Create a JDBC connection to an Oracle BI EE Server that contains the data you want to display in the ADF page. For more information, see "Creating a BI JDBC Connection".
2. Create an application module with a standard JDBC connection. For more information, see "Initializing the ADF Project".
3. Create an Oracle Business Intelligence view object, pointing to the JDBC connection and using the following logical SQL. Note that Region, District, and Dollars will be added automatically as the attributes of the View Object.

   ```
   "Select Markets.Region, Markets.District, Sales
   Measures.Dollars from Paint"
   ```

4. Add a bind variable mapped to the Dollars attribute.
5. Navigate to the Application Model page of the Create View Object wizard and add an instance of the view object that you created to an application module. In this use case, assume that the application module is named "BIVO_AppModule."

   JDeveloper will automatically create a BIVO_AppModuleDataControl for you. Your View Object will be listed as a collection with all its attributes under this data control. The DataControl will also list the "ExecuteWithParam" operation that will take the Dollars parameter.
6. Drag and drop the view object collection from the data control to the appropriate place on a JSPX page. When the pop-up menu appears, select an appropriate ADF Rich UI component to use for displaying the view object.
7. Select a read-only Rich Table for the display.
8. Create an appropriate input component to prompt the user for the threshold Revenue value.
9. Invoke the ExecuteWithParam operation to pass in the threshold revenue value when the user submits the form and refresh the Rich Table that you created earlier.

2.2.3 Displaying Data from the Online Transaction Processing Schema and Logical SQL Together in an ADF-Rich Component

This use case describes how to display an ADF Rich Table that contains the following columns:

- Customer
- Accounts Payable Days Outstanding
- Lifetime Value
- Customer Rank by YTD Revenue

The "Accounts Payable Days Outstanding" value comes from an online transaction processing (OLTP) system and "Lifetime Value" and "Customer Rank" come from the Oracle Business Intelligence Server. "Customer" is common to both systems.

To display data from the online transaction processing schema and logical SQL together in an ADF-rich component:

1. Create a JDBC connection to an Oracle BI EE Server that contains the data you want to display in the ADF page. For more information, see "Creating a BI JDBC Connection".
2. Create an application module with a standard JDBC connection. For more information, see "Initializing the ADF Project".
3. Create a regular view object within the OLTP system and uses the appropriate SQL/Entity Objects to expose the Customer and Accounts Payable Days Outstanding attributes. In this use case, assume that the view object is named OLTP_VO.
4. Using the JDBC connection to the Oracle BI EE Server that you created in Step 1 of this procedure, create an Oracle Business Intelligence view object, using the following logical SQL:

   ```
   "Select Customer, Lifetime_Value, Rank(YTD Revenue) from
   CustomerSubjectArea"
   ```

Customer, Lifetime Value and Customer Rank by YTD Revenue will be automatically added as the attributes of this view object. In this use case, assume that the view is named BI_VO.

5. Create a view link with the Customer attribute of the OLTP_VO as the source attribute, and the Customer attribute of the BI_VO as the destination attribute.
6. Navigate to the Java step of the OLTP_VO and generate the Java classes for the view object. Also add attributes to the OLTP_VO for "Lifetime_Value" and "RankByRevenue".
7. Edit the Java class for the OLTP_VO (OLTP_VOViewImpl). There will be an accessor method for the each attribute added in the previous step. Edit the implementation of the accessors to return the corresponding values from the Oracle Business Intelligence view object. The business intelligence view object is available to the OLTP_VO View object through the getViewObject method. For example, the code for the Lifetime_Value accessor will look similar to the following:

   ```
   getViewObject().getCurrentRow().getAttribute("Lifetime_
   Value")
   ```
8. Confirm that OLTP_VO is a part of the ApplicationModule. The OLTP_VO will have a collection of all of the attributes (Customer, Accounts Payable Days Outstanding, Lifetime Value", Customer Rank by YTD Revenue).
9. Drag and drop the view object collection from the data control to the appropriate place on a JSPX page. When the pop-up menu appears, select the ADF Rich UI component to use for displaying the view object.
10. Select a read-only Rich Table for the display.

2.2.4 Creating a Master-Detail Relationship Between an Oracle Business Intelligence View Object and an Online Transaction Processing View Object

In this use case, you create the master-detail relationship in the same way that you create a master-detail relationship between two OLTP view objects. The instructions provided in the following procedure are high-level instructions. You can find complete information about creating a master-detail relationship in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

To create a master-detail relationship between an Oracle Business Intelligence view object and an online transaction processing view object:

1. Create an OLTP View Object.
2. Create a Oracle Business Intelligence view object.
3. Create a View Link between the OLTP View object and the BI View Object.
4. Create an appropriate ADF View Object for the source and the destination data controls corresponding to the source and destination view objects.

2.3 Creating an Oracle Business Intelligence Logical SQL View Object

Before you can create the Oracle BI Enterprise Edition Logical SQL View Object, you must perform two preparatory steps: creating a BI JDBC connection and initializing the ADF project with a JDBC connection.

After creating and saving the Oracle BI Enterprise Edition Logical SQL View Object, you can modify the view object behavior so that it interacts in a specific way with the ADF application's other view objects.

2.3.1 How to Create an Oracle Business Intelligence Logical SQL View Object

Use the following subprocedures to prepare for and create an Oracle BI Enterprise Edition Logical SQL View Object that you can use to bind Oracle BI EE data to ADF components.

2.3.1.1 Installing the Required Business Intelligence JDeveloper Extensions

Use JDeveloper's Check for Updates wizard to find and install the required Oracle BI EE extensions.

To install the Oracle BI EE extensions:

1. From the JDeveloper Help menu, select **Check for Updates**.
2. Follow the steps in the wizard to locate, download, and install the Oracle BI EE extensions.

2.3.1.2 Creating a BI JDBC Connection

You must create a BI JDBC connection to the Oracle BI EE Server. It is through this connection that the view object will pass the logical SQL statement to the Oracle BI EE Server, and through which the Oracle BI EE Server will pass data back to the view object in the ADF application.

You can create the BI JDBC connection to be available for any ADF application, or in the context of a specific ADF application.

To create a BI JDBC connection:

1. Determine if you want to create the BI JDBC connection for use in all ADF application or for the current ADF application, only, and then navigate to the Create Database Connection dialog.

   To create the connection in an ADF application, open an ADF project, go to the Application Navigator, then to the Application Resources panel, right-click the Connections node, select **New Connection**, and select **Database**. The Create Database Connection dialog is displayed. Confirm that Application Resources is selected in the Create Connection In field.

   To create the connection that is available for all ADF applications, go to the Resource Palette, click the **New** icon, select **New Connection**, and select **Database**. The Create Database Connection dialog is displayed. Confirm that IDE Connections is selected in the **Create Connection In** field.

2. In the **Connection Type** field, select BI (JDBC) and enter the appropriate connection information. For more information, press **F1** or click **Help** from within the Create Database Connection dialog.
3. Click **Test Connection**.

   - If the test succeeds, a "Success!" message displays in the status text area.
   - If the test does not succeed, an error displays. Change the information that you entered in the dialog as needed to correct the error, or check the error content to determine other possible sources of the error.

   The new connection name appears under the appropriate node in the Application Navigator or the Resource Palette.

2.3.1.3 Initializing the ADF Project

If the Oracle Business Intelligence view object that you are creating is the first business component added to the ADF application, you must initialize the project with a standard JDBC connection. You cannot initialize the project with a BI JDBC connection. The BI JDBC connection is specified later when you create the BI View object.

Note: You do not need to initialize a Business Components project. Business Components projects are initialized at creation time.

To initialize the ADF project:

1. Open an ADF project, click **File**, and then select **New**. The New Gallery dialog is displayed.
2. Select the All Technologies tab and in the Categories area, select Business Tier and then select Business Intelligence.
3. Confirm that View Object (Business Intelligence Logical SQL View Object) is highlighted in the **Items** area and click **OK**. The Initialize Business Components Project dialog is displayed.
4. Specify the database connection and SQL flavor your components will use when creating an initial business components project, or when setting project properties. For more information, press **F1** or click **Help** from within the Create Database Connection dialog.
5. Click **OK**. The Create View Object wizard is displayed.

2.3.1.4 Creating the Business Intelligence View Object

Use the following procedure to create and save the Oracle Business Intelligence Logical SQL view object using the Create View Object wizard. For more information about a specific dialog within the Create View Object wizard, press **F1** or click **Help** from within the dialog upon which you are working.

After you have created and saved the view object, you can then access the saved view object and add the following items to the view object: alternate keys, client interface specifications, client row interface specifications, view accessors, and user interface hints.

1. Open an ADF project, click **File**, and then select **New**. The New Gallery dialog is displayed.

   Note: If you have not yet initialized the ADF project, JDeveloper requires you to do so before you can create a business intelligence view object. For more information, see "Initializing the ADF Project".

2. Select the All Technologies tab and in the Categories area, select Business Tier and then select Business Intelligence, confirm that View Object (Business Intelligence Logical SQL View Object) is highlighted in the **Items** area. Click **OK**.
3. In the Name screen, enter a name for the view object and specify the package into which you want to place the view object. You can use the **Property Set** field to select an existing property set for the view object. Click **Next**. The Connection screen displays.

Note: After you save the view object, you cannot modify its name.

4. Select the BI JDBC connection that will run the logical SQL and retrieve the business intelligence data. The JDBC connection that was initialized for the project will default to this connection, but you must choose a BI JDBC connection from the list of available connections or click the **Add** button to create a new BI JDBC connection for the project.
5. Click **Next**. The Logical SQL screen displays.
6. Type the logical SQL to execute against the Oracle BI EE Server. You can use most of the logical SQL select statements that are supported by the Oracle BI EE Presentation Server. The items in the select list are automatically available as attributes of the business intelligence view object. For example, if the logical SQL is `"Select Markets.Region saw_0, Measures.Dollars saw_1 from Paint"`, the corresponding view object will have two attributes called "region" and "dollars."

   Note: The Insert, Update, and Delete logical SQL statements are not supported by the business intelligence view object. For more information about and a list of the logical SQL statements that you can use with a business intelligence view object, see the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

7. Click **Next**. The Bind Variables screen displays.
8. Add or create any necessary variables. Any bind variables that you create are available as parameters on the data controls that include this view object. For more information about bind variables, press F1 or click Help from within the dialog.
9. Click **Next**. The Java screen displays.
10. Generate any necessary Java files to extend or customize the functionality provided by the business intelligence view object and click **Next**. The Application Model screen appears.

    For more information about generating custom Java classes for the view object, press **F1** or click **Help** from within the dialog.
11. Add an instance of the business intelligence view object to a new or saved Application Module. Note that if you do not add an application module, the data control for the view object will not be automatically generated.
12. Click **Next**. The Summary screen displays.
13. Review the summary and confirm your selections. After you have confirmed that the selections are correct, click **Finish** and JDeveloper creates the Oracle Business Intelligence View Object.

2.3.1.5 Modifying the Oracle Business Intelligence Logical SQL View Object's Behavior

Use the following procedure to modify the view object's behavior and define any additional behaviors, such as client interface specifications, client row interface specifications, view accessors, and user interface hints. The behaviors that you specify will determine how the Oracle Business Intelligence Logical SQL view object interacts with the application's other ADF view objects.

For more information about a specific dialog within the View Object Editor, press **F1** or click **Help** from within the editor.

1. To locate and open the view object, click **File** and then click **Open**. The Open dialog is displayed, where you can browse to and select the view object. The Overview Editor is displayed.
2. Click the Connection node to change the view object's BI JDBC connection information, or add or switch to another BI JDBC connection.
3. Click the Logical SQL node to modify the view object's logical SQL and to add named bind variables to the SQL query.
4. Click the Java node to generate Java files for the view object and its view row set. The generated files allow the client to call methods on the object remotely, generates a service data object that wraps a view row, and allows the view object to be exposed in a web service interface.
5. Click the View Accessors node to define the view accessors to point from a base entity object attribute or view object attribute to a source view row set.
6. Click the List UI Hints node to specify a default list of values (LOV) hint when you want the LOV-enabled attributes of other view objects to inherit the list UI hints from the current view object.

3

Integrating Oracle Business Intelligence Presentation Services With Oracle WebCenter

This chapter explains how you use JDeveloper to create and configure a WebCenter application that integrates with the Oracle Business Intelligence Presentation Services catalog. At runtime, users can add business intelligence objects to their WebCenter Spaces pages.

This chapter includes the following sections:

- "Introduction to the WebCenter and Oracle BI Presentation Services Integration"
- "Creating a WebCenter Application that Integrates with Oracle BI Presentation Services"

3.1 Introduction to the WebCenter and Oracle BI Presentation Services Integration

You can use JDeveloper to create WebCenter applications that integrate with Oracle Business Intelligence Presentation Services. At runtime, these applications include the Presentation Services catalog in the WebCenter Resource catalog. Users can then browse for and add business intelligence analyses, dashboard pages, dashboards, or scorecard components (strategy maps, strategy trees, KPI watchlists, cause and effect maps, and custom views) to their WebCenter Spaces pages. Any filters, prompts, and actions links included in the business intelligence objects will work within the WebCenter Spaces pages.

At runtime, WebCenter Spaces users can expand and browse the Presentation Services catalog's folders to view an analysis' views. The following view types display in the Presentation Services catalog: table, pivot table, chart, funnel chart, gauge, narrative, ticker and title. The following view types do not display in the Presentation Services catalog: view selector, column selector, logical SQL, and no-results view.

Users can also browse the dashboard folder for the pages associated with the dashboard; however, users cannot browse within the dashboard pages to see their components (for example, any analyses embedded in the dashboard). Users cannot include entire Scorecards (only a Scorecard's components) in their WebCenter Spaces pages.

3.2 Creating a WebCenter Application that Integrates with Oracle BI Presentation Services

To create a WebCenter application that integrates with Oracle Business Intelligence Presentation Services, you must have installed Oracle JDeveloper 11*g* Release 1 (11.1.1) and the required Oracle BI EE and WebCenter extensions. To properly create the WebCenter application, you must configure the library settings, update the web.xml and default-catalog.xml files, and have properly configured security. For more information about configuring security, see "How to Configure SSL and Create an Oracle BI EE Presentation Services Connection".

3.2.1 How to Install the Required JDeveloper Extensions

Use the following procedures to confirm that you have the required Oracle JDeveloper 11g Release 1 (11.1.1) extensions and if necessary, install the extensions.

3.2.1.1 Checking for Installed JDeveloper Extensions

Use this procedure if you are not sure whether you have the required WebCenter and Oracle BI EE extensions.

1. In JDeveloper, click the **Help** menu and click **About**. The "About Oracle JDeveloper 11g" dialog displays.
2. Click the **Extensions** tab to view a list of the installed extensions. To sort the list, click either the **Name** or **Identifier** column heading.

3.2.1.2 Installing the Required JDeveloper Extensions

Use this procedure to install the require extensions.

1. From the JDeveloper **Help** menu, select **Check for Updates**.
2. Follow the steps in the wizard to locate, download, and install the WebCenter and Oracle BI EE extensions.

3.2.2 How to Create a WebCenter Application that Integrates with Oracle BI Presentation Services

Use the following procedure to create or modify an Oracle WebCenter application that can access the Oracle Business Intelligence Presentation Services catalog at runtime.

3.2.2.1 Creating a WebCenter Application

Use the following procedure to create WebCenter application and JSF page.

To create a WebCenter application:

1. In Oracle JDeveloper, click **File** and select **New**. The New Gallery dialog displays.
2. In the Categories field’s tree structure, locate the General branch and underneath it, find and click **Applications**. A list with descriptions of the available application types displays in the Items field. From this list, select WebCenter Application and click **OK**. The Create New WebCenter Application wizard displays.
3. Use the wizard to set up and create the WebCenter application. For more information about setting up the application, click the **Help** button to access the online help.
4. Click **Finish**.

5. In the application's Project pane, locate, highlight, and right-mouse click ViewController. From the list of options, select **New**. The New Gallery dialog displays.

6. In the Categories field's tree structure, locate the Web Tier. A list with the descriptions of the available options display in the Items field. From this list, select JSF Page and click **OK**. The Create JSF Page dialog displays.

7. Create the JSF page, making sure that the **Create an XML Document (*.jspx)** field is selected. For more information about setting up the JSF page, click the **Help** button to access the online help. Click **OK**. The newly created JSPX page displays.

8. In the Component Palette pane, locate the options list and select Oracle Composer. The field updates to display the corresponding components, layouts, and operations.

9. From the Common Components category, select **Change Mode Link** and drag and drop it in the following location within the <your_page_name>.jspx - Structure tab:

   ```
   jsp.root > f:view > af:form > pe:changeModeLink
   ```

10. From the Common Components category, select **Page Customizations** and confirm that Oracle JDeveloper adds it *after* the pe:changeModeLink item that you added in the above step. It, along with its subordinate items (cust:panelCustomizable and Page Customizable facets), should be located in the following location within the <your_page_name>.jspx - Structure tab:

    ```
    jsp.root > f:view > af:form > pe:pageCustomizable
    ```

11. Save the application.

3.2.2.2 Configuring the Application to Display the Oracle BI EE Presentation Catalog

Use this procedure to add the Presentation Catalog to the WebCenter application's folder list. At runtime, the catalog displays in the WebCenter Resource Catalog where the users can browse to business intelligence objects and add them to their WebCenter Spaces page.

To display the Presentation Catalog in the WebCenter application's folder list:

1. In the WebCenter application that you created with Oracle JDeveloper, locate the Project pane, select **ViewController**, right-click, and select **Project Properties**. The Project Properties dialog displays.

2. In the available properties pane, select Technology Scope and from the Available field, select BI ADF Components, and click the **Add (>)** button to add it to the project's technology scope.

3. In the available properties pane, select Libraries and Classpath, and click the **Add** button. The Add Library dialog displays.

4. Add the following libraries to the project, and click **OK**.

   - BI ADF
   - BI ADF Taskflow
   - BI SOAP Connection Runtime
   - BI View Object Runtime

5. Save the application.

6. Locate and open the following default resource catalog file that was created when you saved the WebCenter application:

   <your_application_home>\mds\oracle\adf\rc\metadata\default-catalog.xml

7. Add the following `<customFolder>` entry under the `<catalogDefinition> - ><contents>` folder:

```
<customFolder id="oracleBIContent"
    name="Oracle BI"
    description="Oracle BI Presentation Services Content"
    factoryClass="oracle.bi.presentation.rescat.
      BICatalogInitialContextFactory"/>
```

8. In the application's Project pane, locate and open the web.xml file.

9. Update the file to include the BIRegionServlet definition by adding the BiRegionServlet definition:

```
<servlet>
    <servlet-name>biregionservlet</servlet-name>
    <servlet-class>oracle.bi.presentation.servlet.
      BIRegionServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>biregionservlet</servlet-name>
    <url-pattern>/BIProxy/*</url-pattern>
</servlet-mapping>
```

   Make sure that the filter mapping of BIRegionFilter displays before the filter-mapping definition. For example:

```
<filter-mapping>
    <filter-name>BIRegionFilter</filter-name>
    <servlet-name>Faces Servlet</servlet-name>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>

<filter-mapping>
    <filter-name>adfBindings</filter-name>
    <servlet-name>Faces Servlet</servlet-name>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

3.2.2.3 Enabling the BI ADF Taskflow Permission

Use the following procedure to enable the BI ADF taskflow permissions in your application's security settings.

To enable the BI ADF taskflow permissions:

1. Locate and open the jazn-data.xml file. This file resides in the following location:

   <your_application_root>\src\META-INF\jazn-data.xml

2. Add the following element to this file:

```
<permission>
    <class>oracle.adf.controller.security.TaskFlowPermission</class>
    <name>/WEB-INF/oracle/bi/presentation/taskflow/
```

```
            task-flow-definition.xml#task-flow-definition</name>
        <actions>customize,personalize,view</actions>
    </permission>
```

3.2.3 Testing the WebCenter Application

Use the following procedures to test the WebCenter application. You can test the application by performing the following procedures in WebCenter Spaces: add Oracle BI Presentation Services catalog content, modifying a report or dashboard's prompt values, and confirm and modify a business intelligence object's initialization parameters.

3.2.3.1 Adding Business Intelligence Content at Runtime

Use this procedure to test the WebCenter application by adding business intelligence content to the WebCenter application.

1. Run the .jspx page that you created.
2. In the running page, click the **Edit** link to enter the edit mode.
3. Click **Add Content**. The WebCenter Resource Catalog displays.
4. Browse to and open the Oracle BI folder.
5. Click the host machine to which you want to connect. The contents of the catalog display.
6. Browse to and click the **Add** link for the object that you want to add to the .jspx page.
7. Confirm that the application correctly added the object.

3.2.3.2 Modifying a Business Intelligence Object's Prompt Values

Use this procedure to test the WebCenter application by changing an analysis or dashboard's filter or prompt values.

1. Run the .jspx page that you created.
2. In the running page, click the **Edit** link to enter the edit mode.
3. Add an analysis or dashboard that contains a filter or prompt. For more information about adding business intelligence content to the .jspx page, see "Adding Business Intelligence Content at Runtime".
4. Click **Configure**. The "Editing Page dialog" displays.
5. Modify the values and click **Save**.
6. Confirm that the application correctly applied the modified prompt values.

3.2.3.3 Modifying a Business Intelligence Task Flow's Initialization Parameters

Use the following procedure to test the business intelligence task flow's initialization parameters.

1. Run the .jspx page that you created.
2. In the running page, click the **Edit** link to enter the task flow's edit mode.
3. Add a business intelligence object. For more information about adding business intelligence content to the .jspx page, see "Adding Business Intelligence Content at Runtime".

4. Locate the business intelligence object and click **Edit**. The "Component Properties dialog" displays.
5. Go to the **Parameters** tab and modify the object's parameters. Click **OK**.
6. Confirm that the application correctly applied the modified prompt values.

3.2.4 Post-Deployment Connection Configuration

After an application has been deployed, you can use Oracle Enterprise Manager to add a new Oracle BI EE Presentation Services connection to the application, or to modify the application's existing Oracle BI EE Presentation Services connection.

For more information about modifying these connections, see "Adding or Modifying a Presentation Services Connection After Deployment".

3.2.5 Securing Oracle BI EE Objects Included in a WebCenter Application

Impersonate User allows you to secure a WebCenter application that integrates with Oracle BI EE Presentation Services. For more information about Impersonate User and how to implement it to secure your application, see "How to Create and Use Impersonate User".

Glossary

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 Business Intelligence catalog objects in ADF Applications. This component uses a SOAP connection to access the Oracle BI Presentation Catalog.

Admin Server

Is part of the WebLogic domain, and runs the processes that manage Oracle Business Intelligence components. The Admin Server contains the Oracle WebLogic Server Administration Console, and Fusion Middleware Control.

See also **Fusion Middleware Control** and **Managed Server**.

Administration Tool

See **Oracle BI Administration Tool**.

agent

Enables you to automate your business processes. You can use them 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 (e-mail, 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. Alias tables can be used to set up multiple tables, each with different keys, names, or joins, when a single physical table needs to 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 containing 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 domain

Contains configurable System components (the coreapplication) and Java components (the WebLogic domain), and also includes the Web-based management tools and applications that utilize 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

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 and 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.

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 will return. Together with filters and selection steps, columns determine what analyses will 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 allows you to build specific value prompts on a data column to either stand 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.

condition

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)**.

connection 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 pre-delivered action that utilizes 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 allow 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 will display in all analyses 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 a number of forms and optional arguments that can be used 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 only supported for Oracle Database data sources.

dimension

A hierarchical organization of logical columns (attributes). One or more logical dimension tables may be associated with at most one dimension.

A dimension may 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**.

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 Oracle BI Presentation Services user interface object that contains links and options that allow the user to quickly begin a task or locate a specific object within the Presentation Catalog. The global header always displays in the Presentation Services user interface, thus allowing 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 your dashboard or external Web site. It has a number of forms and optional arguments that can be used 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 allow 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 will be executed to initialize or refresh the variables 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**.

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 stored in the 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)**.

level

See **hierarchy level**.

logical display folder

Folders used to organize objects in the Business Model and Mapping layer. 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, as well as 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 Admin Server for all of its configuration and deployment information.

See also **Admin Server** and **Fusion Middleware Control**.

measure column

A column that can change for each record and can be added up or aggregated in some way. 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.

mission statement

A statement in a scorecard that specifies the key business goals and priorities that are required to achieve your vision.

See also **Oracle Scorecard and Strategy Management** and **vision statement**.

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 non-relational 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, 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 server**.

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

A file that stores Oracle Business Intelligence metadata. The metadata defines logical schemas, physical schemas, physical-to-logical mappings, aggregate table navigation, and other constructs. The repository file has an extension of .rpd. Oracle BI repositories 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. This XML file version of the repository can be used to programmatically modify the metadata. The 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.

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 allow 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 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.

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 will not be computed by the Oracle BI Server but will instead be passed to another data source. Enables advanced users to leverage data source features and functions without the need to modify the Oracle BI repository.

permissions

Specify which users can access an object, as well as 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 a 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**.

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 allows 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**.

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**.

snowflake schema

A dimensional schema where one or more of the dimensions are partially or completely normalized.

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 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. Oracle Business Intelligence supports standard SQL-92 with several value-added proprietary extensions.

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.

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.

unbalanced hierarchy

A hierarchy where the leaves do not have the same depth. For example, an organization may choose to have data for the current month at the day level, data for the previous 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

Allows 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 allows 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 `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**.

WebLogic 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