Oracle® Fusion Middleware

Developer's Guide for Oracle Business Intelligence Enterprise Edition

11*g* Release 1 (11.1.1)

E10545-05

July 2014

Explains how to incorporate Oracle Business Intelligence Enterprise Edition functionality into applications built using Oracle JDeveloper, Oracle Application Development Framework (ADF), and Oracle WebCenter.

ORACLE®

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

E10545-05

Contents

2   Improving The Performance of ADF Applications That Contain Oracle BI EE Objects

3   Adding BI Composer to ADF Applications

4   Using the Oracle BI EE Logical SQL View Object

Preface

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

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

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

This guide contains 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

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

Access to Oracle Support

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

Related Documentation and Other Resources

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

In addition:

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

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

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/technetwork/middleware/ias/downloads/fusion-requirements-100147.html

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

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

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 preface describes new features and upgrade considerations in Oracle Business Intelligence Enterprise Edition 11*g* Release 1 (11.1.1).

This preface contains the following topics:

- "New Features for Oracle BI 11g Release 1 (11.1.1.9)"
- "New Features for Oracle BI 11g Release 1 (11.1.1.7)"
- "New Features for Oracle BI 11g Release 1 (11.1.1.6)"
- "New Features for Oracle BI EE 11g Release 1 (11.1.1.5)"
- "New Features for Oracle BI EE 11g Release 1 (11.1.1.3)"

New Features for Oracle BI 11*g* Release 1 (11.1.1.9)

There are no new features in Oracle BI EE 11*g* Release 1 (11.1.1.9).

New Features for Oracle BI 11*g* Release 1 (11.1.1.7)

There are no new features in Oracle BI EE 11*g* Release 1 (11.1.1.7).

New Features for Oracle BI 11*g* Release 1 (11.1.1.6)

This section describes new features for Oracle BI EE 11*g* Release 1 (11.1.1.6). It contains the following topics:

- New Features for 11.1.1.6.0
- New Features for 11.1.1.6.2

New Features for 11.1.1.6.0

There are no new features for Oracle BI EE 11*g* Release 1 (11.1.1.6.0).

New Features for 11.1.1.6.2

There are no new features for Oracle BI EE 11*g* Release 1 (11.1.1.6.2).

New Features for Oracle BI EE 11*g* Release 1 (11.1.1.5)

New features in Oracle BI EE 11*g* Release 1 (11.1.1.5) include:

- "Static Resources"

- "WSDL Context"

- "Override Impersonate User Option"

- "BI Composer Configuration"

Static Resources

You can now configure an Oracle BI Presentation Services connection to use static files that reside on the Oracle BI server or another server. The **Static Resources Mode** and **Static Resources Location** fields were added to the BI Presentation Services Connection wizard to allow you to specify your static resources preferences. The practise of using static resources increases the performance of ADF applications that contain Oracle BI EE objects.

For more information, see Chapter 2, "Improving The Performance of ADF Applications That Contain Oracle BI EE Objects."

WSDL Context

This release enables you to specify the WSDL context. The **WSDL Context** field was added to the BI Presentation Services Connection wizard for situations where the System Administrator changes the default context from "analytics-ws" to something else.

For more information, see "How to Configure SSL and Create an Oracle BI EE Presentation Services Connection".

Override Impersonate User Option

The **Perform impersonation when security is enabled** field was added to allow you to disable the impersonate user credentials in cases where ADF security is enabled for your applications.

For more information, see "How to Configure SSL and Create an Oracle BI EE Presentation Services Connection".

BI Composer Configuration

You can add the BI Composer component to ADF applications, WebCenter Portal applications, and WebCenter Portal: Spaces. BI Composer provides functionality that allows end users to quickly and easily create, edit, or view the design of Oracle BI EE analyses from ADF applications, WebCenter Portal applications, and WebCenter Portal: Spaces.

For more information, see Chapter 3, "Adding BI Composer to ADF Applications" and Chapter 5, "Integrating Oracle Business Intelligence Presentation Services With an Oracle WebCenter Portal Application".

New Features for Oracle BI EE 11*g* Release 1 (11.1.1.3)

New features in Oracle BI EE 11*g* Release 1 (11.1.1.3) include:

- "Embedding Business Intelligence Objects Into ADF Applications"

- "Oracle BI EE Logical SQL View Object"

- "Integrating the Oracle BI Presentation Catalog With WebCenter Portal Applications"

Embedding Business Intelligence Objects Into ADF Applications

You can add business intelligence objects stored in the Oracle BI Presentation 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 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 4, "Using the Oracle BI EE Logical SQL View Object".

Integrating the Oracle BI Presentation Catalog With WebCenter Portal Applications

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

For information, see Chapter 5, "Integrating Oracle Business Intelligence Presentation Services With an Oracle WebCenter Portal Application".

1

Embedding Business Intelligence Objects in ADF Applications

This chapter describes how to add Oracle BI EE objects such as analyses, dashboards, and Scorecard components to Oracle ADF applications. It describes how to prepare the ADF project to accept objects, wiring the objects to ADF, and using Impersonate User to secure the application.

This chapter includes the following sections:

- Section 1.1, "Overview of Embedding Business Intelligence Objects in Applications"

- Section 1.2, "Business Intelligence Objects That You Can Add to ADF Pages"

- Section 1.3, "Embedded Business Intelligence Objects Use Cases"

- Section 1.4, "Preparing the ADF Project to Accept Business Intelligence Objects"

- Section 1.5, "Creating an Oracle BI EE Presentation Services Connection"

- Section 1.6, "Adding Oracle Business Intelligence Objects to ADF Pages"

- Section 1.7, "Passing Business Intelligence Content with the Oracle BI EE Contextual Event Action"

- Section 1.8, "Adding or Modifying a Presentation Services Connection After Deployment"

- Section 1.9, "Credentials for Connecting to the Oracle BI Presentation Catalog"

- Section 1.10, "Deploying Credentials to a Standalone WebLogic Server"

1.1 Overview of Embedding Business Intelligence Objects in Applications

Embedding business intelligence objects allows you to access the Oracle BI Presentation 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 an object to an ADF page, that page contains a reference to the object and does not contain a copy of the object. When the object is modified and saved to the Oracle BI Presentation Catalog, any changes display 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 4, "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 Oracle BI Presentation 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 and *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 analysis for different regions with conditional formatting and take an action if the sales is trending below a minimum threshold. This analysis 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 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

Oracle JDeveloper offers the following Oracle BI EE extensions:

- Business Intelligence ADF Task Flow
- Business Intelligence ADF View Regions
- Business Intelligence Composer
- Business Intelligence Logical SQL View Objects
- Business Intelligence Soap Connection

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

1. In JDeveloper, click the **Help** menu and click **About**. The "About Oracle JDeveloper 11g" dialog displays where you can confirm the release number.

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.

   Note that all Oracle BI EE extensions contain the "bi-" prefix. If no extensions begin with the "bi-" prefix, then you do not have the required extensions.

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 Oracle BI EE extensions. If you create a WebCenter Portal application that integrates with Oracle BI EE, then you need to also install the WebCenter Portal framework.

   For a list of the available Oracle BI EE extensions, see "Checking for Installed JDeveloper Extensions".

1.4.1.3 About Oracle BI EE Personalizations in ADF Applications

Personalizations for dashboards, dashboard pages, analyses, 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

- Standalone Scorecard components

1.4.1.4 Adding the MDS Libraries and Enabling MDS Customizations

To save the site and user application personalizations such as prompt values, 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 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. 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 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 JDeveloper. 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 is `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 installations of Presentation Services. Depending upon where you created and saved the connection, the connection name displays 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 Oracle BI Presentation 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.

You must build this connection only one time 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 during or 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. Use the following substeps to create the connection to be available for any JDeveloper application.

a. Navigate to the Resource Palette, click the **New** button.

b. From the list, 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.

2. Use the following substeps to create the connection to be available for the current application, only:

a. 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.

b. Right click on the **Connections** folder and select **New Connections**.

c. Select **BI Presentation Services**. The Create BI Presentation Services Connection wizard displays.

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

4. Click **Next**.

5. 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 lowercase.

- **WSDL Context**– Enter the context to which you want to deploy the WSDL application. Modify this field only if the System Administrator has changed the default context from "analytics-ws" to something else. For example, mycompanyname-ws. The SOAP connection uses the context specified in this field.

- **Static Resources Mode**– Specify from where to retrieve static resources.

  Select **Auto** to direct all requests to Oracle BI Presentation Services rather than to the Oracle BI proxy. Selecting **Auto** results in a significant increase in performance. Note that **Auto** mode provides the best performance if Oracle WebLogic Server resides behind Oracle HTTP Server or Apache HTTP Server and you are bypassing Oracle WebLogic Server for serving static files.

  Select **Manual** to direct requests to another location. If you select **Manual**, you must specify a URL in the **Static Resources Location** field.

- **Static Resources Location**– (Optional) If you selected **Manual** in the **Static Resources Mode** field, then enter the location of the deployed static resources if they are not located within the Oracle BI installation. The URL format must be http://*host*:*port*/analytics. Examples of static resources are .css and .javascript. Or, leave this field blank if Oracle BI resides behind a firewall but the ports to the ADF instance are not behind a firewall. For more information about setting up static resources, see "Improving Oracle BI Web Client Performance" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

6. Click **Next**.

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

Note: If ADF security is enabled for the application, you must change the user name and password to Impersonate User credentials before you deploy the application. For more information, see "How to Create and Use Impersonate User".

> **Note:** One of the following scenarios must be configured depending on whether you use a shared LDAP or a non-central identity store:
>
> - Shared LDAP identity store (for example OIM).
>
>   Must be set up between the JDeveloper WebLogic server (if using JDeveloper's integrated Oracle WebLogic Server) or your application's WebLogic Server, and Oracle BI Server.
>
> - Non-central identity store (for example, a local WebLogic identity store in the BI instance.
>
>   Each user who logs in to your application must have a corresponding identical user set up in the BI Presentation Server with appropriate groups.

8. If ADF security is enabled for your application and you want to use the Impersonate User credentials, leave this field set to **True**.

   If ADF security is enabled for your application, and you do not want to use the Impersonate User credentials, set the **Perform impersonation when security is enabled** field to **False** to disable the Impersonate User credentials. If you set this field to **False**, you must also specify the necessary credentials in the **User Name** and **Password** fields on this screen (used instead of the SSO user). For more information about credentials, see "Credentials for Connecting to the Oracle BI Presentation Catalog". This setting is not recommended.

   If ADF security is not enabled for your application, you do not have to modify this field. This field is only applied to applications where ADF security is enabled.

9. Click **Next**

10. Click **Test Connection** to verify that the connection information is correct. If your test connection fails, JDeveloper provides you with error messages that help you troubleshoot the problem.

11. 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.mycompany.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 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 Oracle BI Presentation Catalog, the 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 are 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 ADF Contextual Event action is delivered with Oracle BI EE and uses the Oracle BI EE Action Framework. Before the analysis designer can include this action type, the administrator must enable the ADF Contextual Event action to display in the New Action Link menu within Presentation Services. To enable this action type, the administrator must modify the Oracle BI EE instanceconfig.xml file to include the following parameter:

```
<ActionLinks>
    <EnableADFContextualEvent>True</EnableADFContextualEvent>
</ActionLinks>
```

For information about modifying instanceconfig.xml, see "Making Advanced Configuration Changes for Presentation Services" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

At design time, the analysis designer determines which column should contain the Oracle BI EE ADF Contextual Event action, and from the New Action Link menu, selects **ADF Contextual Event**. At runtime, the Oracle BI EE ADF 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 ADF 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 ADF 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 ADF 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 automatically promotes 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 ADF 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
```

1.7.3 How to Detect the Oracle BI EE ADF Contextual Event

Use the following procedure to add event details to the page definition containing the ADF object that will detect the Oracle BI EE ADF contextual event. For more information about page definition files, see *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

1.7.3.1 Detecting the ADF Contextual Event

1. In JDeveloper, locate and open the <*page_name*>PageDef.xml file.

2. Locate the </*pageDefinition*> tag and after it add the following event details, specifying the event name and consumer handler name:

```
<eventMap xmlns="http://xmlns.mycompany.com/adfm/event">
  <event name="Event Name">
   <producer region="*">
    <consumer handler="Handler Name">
     <parameters>
      <parameter name="qdr" value="${payLoad}"/>
     <parameters>
    </consumer>
   </producer>
```

```
    </event>
  </eventMap>
```

3. Locate the *</bindings>* tag and before it add the following method action details:

```
<methodAction id="setOutput" InstanceName="Output.dataProvider"
        DataControl="Output" RequiresUpdateModel="true"
        Action="invokeMethod" MethodName="setOutput"
        IsViewObjectMethod="false"

ReturnName="Output.methodResults.setOutput_Output_dataProvider_setOutput_
result">
      <NamedData NDName="output" NDType="java.lang.String"/>
</methodAction>
```

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 to the WebCenter Portal: 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 Fusion Middleware Control.

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 Adding or Modifying a Presentation Services Connection

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

1. Open Fusion Middleware Control 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 Portal: 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 Fusion Middleware Control 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 Presentation Catalog

At design time, you need to specify credentials to connect to the Oracle BI Presentation Catalog. These credentials are used to retrieve the list of business intelligence objects (for example, analyses, dashboards, and scorecard components) from the Oracle BI Presentation Catalog.

This process ensures that the login to the Presentation Server equals 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. ADF security must be enabled for your application before you can apply the impersonate user credentials to the Oracle BI EE Presentation Services connection.

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, ensure 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 BIImpersonateUser

User the following procedure to use Fusion Middleware Control to grant permissions to BIImpersonateUser.

1. Open Fusion Middleware Control for your Oracle BI EE instance.

2. Locate and select bifoundation_domain. The bifoundation_domain pane displays.

3. In the bifoundation_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 **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

1.10 Deploying Credentials to a Standalone WebLogic Server

For information about manual configuration and packaging recommended for Java EE applications that use OPSS, see Chapter 21 Configuring Java EE Applications to Use OPSS in *Oracle Fusion Middleware Application Security Guide*.

Credentials packaged at the application level are migrated to the domain level credential store as specified in the listener configured in weblogic-application.xml. For example:

```
<listener>
  <listener-class>oracle.security.jps.wls.listeners.Jps
   ApplicationLifecycleListener</listener-class>
</listener>
```

By default in development mode, the credential migration parameter is set to OVERWRITE in weblogic-application.xml. Note that you cannot use OVERWRITE in production mode. The following is an example of how the credential migration parameter it set for development mode:

```
<application-param>
    <param-name>jps.credstore.migration</param-name>
    <param-value>OVERWRITE</param-value>
</application-param>
```

1.10.1 Migrating Credentials

Use this procedure to automatically migrate credentials from the application level to the domain level.

For more information about the credential migration parameter, see "Credential Parameter Configuration According to Behavior" in *Oracle Fusion Middleware Application Security Guide*.

To migrate credentials

Depending upon the mode in which you are working (development mode or production mode), use one of the following options to automatically migrate credentials from the application level to the domain level.

- While in *development mode*, confirm that WebLogic Server was started with the migrate overwriting credentials operation, jps.app.credential.overwrite set to true. For example:

  ```
  jps.app.credential.overwrite.allowed=true
  ```

- While in *production mode*, open weblogic-application.xml and change the jps.credstore.migration parameter to MERGE. Save the file, repackage the EAR file, and then deploy the file to the standalone WebLogic Server.

  Note that completing this task will migrate the application level credentials to the domain level store only if credentials do not exist in the domain level store. For example, suppose you packaged an application with the following database credentials:

  - User name – User1

  - Password – Pass1

  These credentials (User 1 and Pass1) are migrated because it is the first time credentials are being migrated.

  If the password is changed from Pass1 to Pass2, then Pass1 will not be overwritten with Pass2 because jps.credstore.migration parameter is set to MERGE.

2

Improving The Performance of ADF Applications That Contain Oracle BI EE Objects

This chapter describes tuning procedures, such as adding and setting timeouts and using static resources, that you can perform to increase the performance of Oracle ADF applications that contain Oracle BI EE objects.

This chapter includes the following sections:

- Section 2.1, "Setting an Oracle BI Server Timeout"
- Section 2.2, "Adding a Session Timeout"
- Section 2.3, "Using Static Resources"

2.1 Setting an Oracle BI Server Timeout

You can improve system performance by passing properties to Oracle WebLogic Server that specify how long a SOAP request waits for a response from the Oracle BI Server before timing out.

2.1.1 How to Set the Oracle BI Server Timeout

Pass the following properties to Oracle WebLogic Server to tune how long a SOAP request waits for a response from Oracle BI Server before timing out. Note that the initial recommended setting is approximately 60 seconds. The value that you specify should always be less than what is configured for stuck thread detection in WebLogic Server.

For more information about configuring stuck thread detection, see "Tuning the Stuck Thread Detection Behavior" in *Oracle Fusion Middleware Performance and Tuning for Oracle WebLogic Server*.

For more information, see "Examples of Timeout Settings".

- `-Dsun.net.client.defaultReadTimeout=`*timeout_in_milliseconds*
- `-Dsun.net.client.defaultConnectTimeout=`*timeout_in_milliseconds*

2.1.2 Examples of Timeout Settings

The following examples show how to set the timeout value to 60 seconds.

For Windows:

`commEnv.cmd`

```
set CONFIG_JVM_ARGS=-Dsun.net.client.defaultReadTimeout=60000
-Dsun.net.client.defaultConnectTimeout=6000
```

For Linux:

```
commEnv.s
setenv CONFIG_JVM_ARGS -Dsun.net.client.defaultReadTimeout=60000
-Dsun.net.client.defaultConnectTimeout=60
```

2.2 Adding a Session Timeout

The Oracle BI ADF integration code supports automatically closing sessions when the ADF HTTP session is closed or times out.

2.2.1 How to Add a Session Timeout

Use the following procedure to modify the application's weblogic.xml so that the application automatically closes.

1. In JDeveloper, go to the Application Navigator, open the Projects tab, navigate to the project's WEB-INF subdirectory, and open the weblogic.xml document.

2. Add the following to the document:

```
<listener>
  <listener-class>oracle.bi.presentation.listener.BIADFServletSessionListener
  </listener-class>
</listener>
```

2.3 Using Static Resources

You can configure an Oracle BI Presentation Services connection to use static resources that reside either on the Oracle BI Server or another server. The practise of specifying static resources, such as .javascript and .css, can significantly increases the performance of ADF applications that contain Oracle BI EE objects.

2.3.1 How to Set Static Resources Preferences

Use the following procedure to specify static resources preferences for a saved Oracle BI Presentation Services connection.

For information about setting up static resources in Oracle BI EE, see "Improving Oracle BI Web Client Performance" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

For information about creating a new connection and specifying static resources, see "Creating an Oracle BI EE Presentation Services Connection".

1. In JDeveloper, go to either the Application Resources tab within the Application Navigator or to the Resource Palette and select the Oracle BI Presentation Services connection to modify.

2. Right-mouse click the selected data connection and select **Properties**. The Edit BI Presentation Services Connection wizard displays.

3. In the BI Presentation Services page, specify the static resources settings. Note the following fields:

    ■   **Static Resources Mode**– Specify from where to retrieve static resources.

Select **Auto** to direct all requests to Oracle BI Presentation Services rather than to the Oracle BI proxy. Selecting **Auto** results in a significant increase in performance. Note that **Auto** mode provides the best performance if Oracle WebLogic Server resides behind Oracle HTTP Server or Apache HTTP Server and you are bypassing Oracle WebLogic Server for serving static files.

Select **Manual** to direct requests to another location. If you select **Manual**, you must specify a URL in the **Static Resources Location** field.

- **Static Resources Location**– (Optional) If you selected **Manual** in the Static Resources Mode field, then enter the location of the deployed static resources if they are not located within the Oracle BI installation. The URL format must be http://host:port/analytics. For example, http://bips:9704/analytics. Examples of static resources are .css and .javascript. Or, leave this field blank if Oracle BI resides behind a firewall but the ports to the ADF instance are not behind a firewall.

4. Click **Finish** to save your changes.

2.3.2 Testing The Static Resources Location

Use the following procedure to test that your static resources location settings (URL) are valid.

1. Create an application that uses the Oracle BI Presentation Services connection in which you specified static resources settings.

2. From the data connection, browse for an analysis and drag and drop it onto a .jspx page.

3. Save the application.

4. Deploy and run the application.

5. Using a tool such as Firebug, validate that requests for .js and .png files are loaded directly from the static resource URL that you specified in the connection. Note that if you receive 404 (or Not Found) error messages that you have specified an invalid URL in your static resources settings.

3

Adding BI Composer to ADF Applications

This chapter describes how to add the BI Composer component to Oracle ADF applications. It describes how to prepare the ADF project to accept BI Composer by extending the domain and embedding the task flow by adding parameters and events.

This chapter includes the following sections:

- Section 3.1, "Overview of Using BI Composer with ADF Application"
- Section 3.2, "Adding BI Composer to an ADF Application"

3.1 Overview of Using BI Composer with ADF Application

You can add the BI Composer functionality to an ADF application. BI Composer is a simple-to-use wizard that allows users to quickly and easily create, edit, or view the design of Oracle BI EE analyses from ADF applications.

When configured, the BI Composer interface allows ADF application users to browse the Oracle BI EE catalog and subject areas, select or modify the analysis' columns, add views (for example, table and graphs), sort and filter data, and apply conditional formatting to data appearing in tables. For more information about how to use the BI Composer interface, see "Using BI Composer to Work with Analyses" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

Adding BI Composer to a WebCenter Portal application or to WebCenter Portal: Spaces requires a slightly different configuration. For information about configuring a WebCenter Portal application or WebCenter Portal: Spaces to use BI Composer, see "Adding BI Composer to a WebCenter Portal Application or to WebCenter Portal: Spaces".

3.2 Adding BI Composer to an ADF Application

To add BI Composer to an ADF application, you must add the BI Composer library reference to the ADF application, apply the domain extension template, and add task flow parameters and events to the ADF application.

3.2.1 Checking for Installed JDeveloper Extensions

Oracle JDeveloper offers the following Oracle BI EE extensions:

- Business Intelligence ADF Task Flow
- Business Intelligence ADF View Regions
- Business Intelligence Composer

- Business Intelligence Logical SQL View Objects

- Business Intelligence Soap Connection

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

1. In JDeveloper, click the **Help** menu and click **About**. The "About Oracle JDeveloper 11g" dialog displays where you can confirm the release number.

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.

   Note that all Oracle BI EE extensions contain the "bi-" prefix. If no extensions begin with the "bi-" prefix, then you do not have the required extensions.

3.2.2 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 Oracle BI EE extensions. If you create a WebCenter Portal application that integrates with Oracle BI EE, then you need to also install the WebCenter Portal framework.

   For a list of the available Oracle BI EE extensions, see"Checking for Installed JDeveloper Extensions".

3.2.3 How to Add the BI Composer Library Reference

Use the following procedure to add the required library reference to the application. Before you can perform this procedure, you must have set up the ADF project to accept Oracle BI EE objects. For more information, see "Preparing the ADF Project to Accept Business Intelligence Objects".

1. Open the application's weblogic.xml.

2. Add the following library reference:

```
<library-ref>
  <library-name>oracle.bi.composer</library-name>
</library-ref>
```

3. Open the application's weblogicapplication.xml document.

4. Add the following library reference:

```
<library-ref>
  <library-name>oracle.bi.jbips</library-name>
</library-ref>
```

3.2.4 How to Extend the Domain

Perform this procedure to apply the domain extension template to deploy the BI Composer implementation libraries.

1. Go to the Oracle BI EE installation directory.

2. Use the following substeps to copy the required extension template and libraries from Oracle BI EE's *Oracle_Home*/and paste them to the target application's *Oracle_Home*/.

a. Go to the Oracle BI EE installation and copy *Oracle_Home*//common/templates/applications/oracle.bicomposer.slib_template_11.1.1.jar. Go to the target application installation and paste the file to *Oracle_Home*//common/templates/applications.

b. Go to the Oracle BI EE installation and copy *Oracle_Home*//modules/oracle.bi.presentation_11.1.1/bi-composer-slib.war. Go to the target application installation and paste the file to *Oracle_Home*//modules/oracle.bi.presentation_11.1.1.

c. Go to the Oracle BI EE installation and copy *Oracle_Home*//modules/oracle.bi.presentation_11.1.1/bi-jbips-slib.ear. Go to the target application install and paste the file to *Oracle_Home*//modules/oracle.bi.presentation_11.1.1.

3. Stop the WebLogic server.

4. Launch the Configuration Wizard.

5. Select **Extend an existing Weblogic domain**, and then click **Next**.

6. Select the domain folder (for example, my_home/user_projects/domains/bifoundation_domain), and then click **Next**.

7. Select **Extend using extension template**, and then use **Browse** to select the following template:

   ORACLE_HOME//common/templates/applications/oracle.bicomposer.slib_template_11.1.1.jar

   When prompted to override the installed template, click **Yes**.

8. Continue clicking **Next** and change any settings if needed.

9. Click **Extend**, and then click **Done**.

10. Start the WebLogic server.

3.2.5 How to Add the Task Flow Parameters and Events

Use the content in this topic to embed the taskflow in an application page.

This topic contains the following subtopics:

- "BI Composer Wizard Editor Task Flow"
- "BI Composer Wizard Edit Events"
- "BI Composer Wizard Viewer Task Flow Parameters"

3.2.5.1 BI Composer Wizard Editor Task Flow

Use the following information to initialize the taskflow with values for parameters based on other elements in the page. For example, you can use these parameters if you have a list of analyses in the page and the selected analysis must be passed as a parameter to the wizard before the user can edit the selected analysis.

The task flow document location is:

```
WEB-INF/oracle/bi/adf/apps/answers/flow/BIReportWizardTF.xml
```

The task flow's ID is:

id: BIReportWizardTF

For more information about adding parameters to a task flow, see Chapter 16: Using Parameters in Task Flows in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

Table 3–1 lists each task flow parameter, its description, type, default value, and if it is required.

Table 3–1   BI Composer Report Wizard Editor Task Flow Parameters

Name	Description	Type	Required?	Default Value
biReportPath	The catalog path for the analysis that is being edited.	String	No	null
biReportContainerPath	The catalog parent folder path for the analysis that is being edited. This parameter is used when selecting an analysis that is a shortcut.	String	No	null
biSubjectArea	The subject area upon which a new analysis is being created.	String	No	null
biConnectionId	The ID of the deployed Oracle BI EE Presentation Services connection.	String	No	bi-default
biCatalogOverwriteAllowed	This flag indicates if the user is allowed to overwrite existing analyses in the catalog.	Boolean	No	false
biReportXML	The analysis' XML to use for editing.	String	No	null
biReportHeight	Dashboard height to be used while rendering a report. For example, for 1024 by 768 pixels resolution, specify biReportHeight as 768px.	String	No	null
biReportWidth	Dashboard width to be used while rendering a report. For example, for 1024 by 768 pixels resolution, specify biReportWidth as 1024px.	String	No	null
biTenantId	Unused	N/A	N/A	N/A
biShowPreviewDefault	This flag indicates if preview is turned on by default.	Boolean	No	false
biShowIncompatibilityMessages	This flag indicates if warning messages should be displayed when opening an analysis that is not compatible with BI Composer.	Boolean	No	true
biReportViewName	Unused	N/A	N/A	N/A
biPromptFilterXML	Unused	N/A	N/A	N/A
biDebugMode	This flag indicates if the icon to generate debug analysis XML is displayed by default.	Boolean	No	false

Table 3–1   (Cont.) BI Composer Report Wizard Editor Task Flow Parameters

Name	Description	Type	Required?	Default Value
biReportLinks	This comma separated value indicates report links to be shown for interactive preview.	String	No	null Valid values are: ■ r Refresh. Displays the Refresh link. ■ f Print. Displays the Print link. ■ d Download. Displays the Export link. ■ - None. Displays no report links. For example, use f,r,d to show all supported links. Null is treated as the default for none.
biViewReportLinks	This parameter specifies which links are to be displayed with an analysis.	String	No	None Valid values are: ■ r Displays the Refresh link. ■ f Displays the Print link. ■ d Displays the Export link. ■ - None. Displays no report links.
biInteractivePreview	This flag indicates if render preview is in interactive mode.	Boolean	No	false

3.2.5.2 BI Composer Wizard Edit Events

Use the following information to add events that perform specific processing from the BI Composer wizard. For example, you can add a Save event that is initiated by the

wizard taskflow after the analysis is edited in the BI Composer wizard. This event refreshes the application's catalog tree.

The task flow document location is:

`WEB-INF/oracle/bi/adf/apps/answers/flow/BIReportWizardTF.xml`

The task flow's ID is:

BIReportWizardTF

For more information about adding parameters to a task flow, see Chapter 16: Using Parameters in Task Flows in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*

Table 3–2 contains each event, its description, and its payload.

Table 3–2   BI Composer Report Wizard Editor Task Flow Events

Event Name	Description	Payload
biAnsReportWizardCancelEvent	Triggered when the user clicks on the wizard's **Cancel** button.	None
biAnsReportWizardSaveEvent	Triggered when the user saves an analysis to the catalog.	Path to the analysis saved in the Catalog.
biAnsReportWizardRefreshCatalogEvent	Triggered when any action, such as rename or delete, is performed that requires external consumers to refresh their catalog tree.	Path to the analysis in the catalog which was the source of the action that requires a catalog refresh.

3.2.5.3 BI Composer Wizard Viewer Task Flow Parameters

Use the following information to initialize the taskflow with values for parameters based on other elements in the page. For example, you can use these parameters if you have a list of analyses in the page and the selected analysis must be passed as a parameter to the viewer so that it can render the analysis.

The task flow document location is:

`WEB-INF/oracle/bi/adf/apps/answers/flow/BIReportViewerTF.xml`

The task flow's ID is:

id: BIReportViewerTF

For more information about adding parameters to a task flow, see Chapter 16: Using Parameters in Task Flows in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*

Table 3–3 lists each task flow parameter, its description, type, default value, and if it is required.

Table 3–3   BI Composer Report Wizard Viewer Task Flow Parameters

Parameter Name	Description	Type	Required?	Default Value
biReportPath	The catalog path for the analysis that is being viewed.	String	No	null

Table 3–3   (Cont.) BI Composer Report Wizard Viewer Task Flow Parameters

Parameter Name	Description	Type	Required?	Default Value
biConnectionId	The ID of the deployed Oracle BI EE Presentation Services connection.	String	No	bi-default
biReportXML	The analysis' XML to use for editing.	String	No	null
biTenantId	Unused	N/A	N/A	N/A
biContentType	The type of content being viewed: "biReportContent" for analyses or "biDashboardContent" for dashboards.	String	No	null
biReportViewName	Unused	N/A	N/A	N/A
biReportLinks	This comma separated value indicates report links to be shown for interactive preview.	String	No	null Valid values are: ■ r Refresh. Displays the Refresh link. ■ f Print. Displays the Print link. ■ d Download. Displays the Export link. ■ - None. Displays no report links. For example, use `f,r,d` to show all supported links. Null is treated as the default for none.

Table 3–3   (Cont.) BI Composer Report Wizard Viewer Task Flow Parameters

Parameter Name	Description	Type	Required?	Default Value
biReportSizing	When set to stretch, this parameter stretches the report or dashboard contents irrespective of biReportHeight and biReportWidth. When set to size, this parameter stretches the report or dashboard contents based on biReportheight and biReportWidth.	String	No	Size

4

Using the Oracle BI EE Logical SQL View Object

This chapter describes how to create and add the Oracle Business Intelligence Logical SQL View Object to bind Oracle BI EE data into an ADF application's components. It describes how to create the required BI JDBC data source and JDBC connection, configure the BI JDBC data source for impersonation, and extend the Oracle Business Intelligence SQL view object.

This chapter includes the following sections:

- Section 4.1, "Introduction to the Oracle BI EE Logical SQL View Object"
- Section 4.2, "Oracle Business Intelligence Logical SQL View Object Use Cases"
- Section 4.3, "Creating an Oracle Business Intelligence Logical SQL View Object"
- Section 4.4, "Creating a BI JDBC Data Source"
- Section 4.5, "Extending the Oracle Business Intelligence Logical SQL View Object"

4.1 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 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 Presentation 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".

4.2 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.

4.2.1 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 are 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 shortcut 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.

4.2.2 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 runtime 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 are 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 automatically creates a BIVO_AppModuleDataControl for you. Your view object is listed as a collection with all its attributes under this data control. The DataControl also lists the "ExecuteWithParam" operation that takes 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 shortcut 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.

4.2.3 Displaying Data from the Online Transaction Processing Schema and Logical SQL 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 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 is 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 is 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 looks similar to the following:

```
getViewObject().getCurrentRow().getAttribute("Lifetime_
Value")
```

8. Confirm that OLTP_VO is a part of the ApplicationModule. The OLTP_VO has 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 shortcut 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.

4.2.4 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 Oracle Business Intelligence 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.

4.3 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.

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

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

4.3.1.1 Checking for Installed JDeveloper Extensions

Oracle JDeveloper offers the following Oracle BI EE extensions:

- Business Intelligence ADF Task Flow

- Business Intelligence ADF View Regions

- Business Intelligence Composer

- Business Intelligence Logical SQL View Objects

- Business Intelligence Soap Connection

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

To check for installed extensions:

1. In JDeveloper, click the **Help** menu and click **About**. The About Oracle JDeveloper 11g dialog displays where you can confirm the release number.

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.

   Note that all Oracle BI EE extensions contain the "bi-" prefix. If no extensions begin with the "bi-" prefix, then you do not have the required extensions.

4.3.1.2 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.

   For a list of the available Oracle BI EE extensions, see "Checking for Installed JDeveloper Extensions".

4.3.1.3 Creating a BI JDBC Connection

You must create a BI JDBC connection to the Oracle BI Server. It is through this connection that the view object passes the logical SQL statement to the Oracle BI

Server, and through which the Oracle BI Server passes 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.

For information about how to create a BI JDBC data source, see "Creating a BI JDBC Data Source".

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. Use the following steps to create the connection in an ADF application.

    a. Open an ADF project and navigate to the Application Navigator.

    b. Go to the Application Resources panel.

    c. Right-click the Connections node and select **New Connection**.

    d. Select **Database**. The Create Database Connection dialog is displayed.

    e. Confirm that Application Resources is selected in the **Create Connection In** field.

2. Use the following steps to create a connection that is available for all ADF applications.

    a. Go to the Resource Palette and click the **New** icon.

    b. Select **New Connection**.

    c. Select **Database**. The Create Database Connection dialog is displayed.

    d. Confirm that IDE Connections is selected in the **Create Connection In** field.

3. 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.

4. 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.

4.3.1.4 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 Oracle Business Intelligence 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 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.

4.3.1.5 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 it:

- alternate keys

- client interface specifications

- client row interface specifications

- view accessors

- user interface hints

To create the view object:

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 an Oracle 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 runs the logical SQL and retrieve the business intelligence data. The JDBC connection that was initialized for the project defaults

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 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 Oracle 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 has two attributes called "region" and "dollars."

> **Note:** The Insert, Update, and Delete logical SQL statements are not supported by the Oracle Business Intelligence view object. For more information about and a list of the logical SQL statements that you can use with an Oracle 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.

   The BI Logical SQL View Object supports JDBC-positional bind variables only.

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 Oracle 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 Oracle 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 is not 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.

4.3.1.6 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 determines 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.

To modify the view object's behavior:

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.

4.4 Creating a BI JDBC Data Source

A BI JDBC data source must exist before you can build a connection to it. It is this data source and connection that allow you to access the Oracle BI Server and pass the logical SQL statement to request data for the view object in the ADF application.

For more information about creating a BI JDBC connection, see "Creating an Oracle Business Intelligence Logical SQL View Object".

4.4.1 How to Create a BI JDBC Data Source in Oracle WebLogic

Use this procedure to access the Oracle WebLogic Server Administration Console to create a BI JDBC data source. Before you begin, ensure that Oracle WebLogic is started.

To create a data source in WebLogic:

1. Access and log in to the WebLogic Server Administration Console for the WebLogic domain in which Oracle BI EE is deployed.

2. In the Domain Structure tree in the left pane, expand **Services** and choose **Data Sources**.

3. Click **New**. You may need to click **Lock & Edit** first to enable the **New** button.

   The Create a New JDBC Data Source wizard displays.

4. On the JDBC Data Source Properties page, follow these steps:

   a. For **Name**, enter the database name of the Oracle Business Intelligence view object in the JDeveloper application. For example, ApplicationOBIS.

   b. For **JNDI Name**, enter the same value you provided for **Name**. Note the following JNDI name format: jdbc/*name from Name field*DS. For example, jdbc/ApplicationOBISDS.

   c. Set the **Database Type** to Other.

      Click **Next**.

   d. Set the **Database Driver** to Other.

    e. Click **Next**.

5. On the Transaction Options page, select **Supports Global Transactions** and select **One-Phase Commit**. Click **Next**.

6. On the Connection Properties page, follow these steps:

    a. For **Database Name**, enter any text string. Note that what you specify in this field is not used for Oracle BI EE.

    b. For **Host Name**, enter the name of the Oracle BI server. For example, sr7619.us.mycompany.com.

    c. For **Port**, enter 9703.

    d. For **Database User Name**, enter a valid Oracle BI EE user name that has been assigned the oracle.bi.server.impersonateUser privilege.

    e. For **Password** and **Confirm Password** enter and then re-enter the password corresponding to the user name that you specified. Click **Next**.

7. On the Test Database Connection page, follow these steps:

    a. For **Driver Class Name**, enter oracle.bi.jdbc.AnaJdbcDriver.

    b. For **URL**, enter jdbc:oraclebi://*Host Name*:9703/. For example, jdbc:oraclebi://sr7619.us.mycompany.com:9703/.

    c. For **Properties**, enter user=*user from Database User Name field* (from above). For example, user=BIImpersonatorUser.

    d. For **Test Table Name**, enter the following: SQL {call NQSGetSQLCatalogs()}

8. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**.

9. Select the server or cluster where you want to deploy the data source. Note that you *must* perform this step before completing the data source configuration.

10. Click **Finish**.

11. Click **Activate Changes**.

4.4.2 Configuring the BI JDBC Data Source for Impersonation

Use this procedure to add a connection string property to the BI JDBC data source. Performing this step enables impersonation. You can skip this step if you do not want the connection to use the currently logged in user.

To configure the BI JDBC data source for impersonation:

1. Access and WebLogic Server Administration Console and locate the BI JDBC data source to configure for impersonation. The Settings page displays.

2. Click the **Connection Pool** tab. Note that you may need to click **Lock & Edit** before you can configure the data source.

3. In the Properties field, create a line and add the following: IMPERSONATE=

4. In the **Initial Capacity** field, enter 1.

5. Click **Save**.

4.5 Extending the Oracle Business Intelligence Logical SQL View Object

If you want to run an Oracle Business Intelligence logical SQL view object from a standalone WebLogic Server, then you must deploy the view object to the standalone WebLogic Server.

To extend the Oracle Business Intelligence logical SQL view object, you must set up a WebLogic domain that supports BI JDBC, prepare the view object, and deploy the application.

4.5.1 How to Set Up a WebLogic Domain that Supports BI JDBC

Use the following procedure to set up a WebLogic domain that supports BI JDBC.

For information about using the Fusion Middleware Configuration Wizard to create or extend a WebLogic domain, see *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard*.

To create a WebLogic domain:

1. Access the configuration wizard by clicking Start, then selecting Programs, then selecting Oracle Fusion Middleware, then selecting WebLogic Server 11gR1, then selecting Tools, and then selecting Configuration Wizard. The Fusion Middleware Configuration Wizard displays.

2. Use the wizard to create a WebLogic domain that supports BI JDBC. Table 4–1 lists the Fusion Middleware Configuration Wizard pages and the corresponding selections you need to make or values you need to specify to correctly set up the WebLogic domain for BI JDBC.

Table 4–1   WebLogic Domain Settings

Wizard Page	Required Selection of Value
Select Domain Source	**Select the following:**
	Oracle BI JDBC - *<your version number>* (oracle_common)
	Select all Oracle JRF options, for example:
	■ Oracle JRF WebServices Asynchronous services - 11.1.1.0 [oracle_common]
	■ Oracle JRF - 11.1.1.0 [oracle_common]
Select Optional Configuration	Select Deployment and Services and then use **Next** to confirm that the required applications are selected.
Target Deployments to Clusters or Servers	Confirm that the following applications are selected:
	■ oracle.bi.adf.model.slib
	■ bijdbc
Creating Domain	Select **Start Admin Server**.
	Or, after you have completed the WebLogic server set up, you can start the server by running `StartWebLogic.sh/cmd` under user_projects/domains/*domain_name*/bin.

4.5.2 How to Prepare the Oracle Business Intelligence Logical SQL View Object and Deploy the Application

Use this procedure to create a connection to the target application server, create the deployment profile, and deploy the application to a standalone WebLogic server. For more information about a specific page used in the following tasks, click the **Help** button on that page.

4.5.2.1 Creating a connection to the target application server

Use the following task to create a connection to the target application server.

To create a connection to the target application server:

1. In JDeveloper, display the Application Server tab by clicking the View menu and selecting Application Server Navigator. The Application Server tab displays.

2. In the Application Server tab, right-click Application Servers and select **New Application Server**. The Create Application Server Connection Wizard - Usage Page displays.

3. On the Usage Page, select **Standalone Server** and click **Next**.

4. Complete the pages on the wizard to set up, test, and save the application server. For information about a specific page, click the page's **Help** button.

    The application server displays in the Application Server tab.

4.5.2.2 Creating the deployment profile for the EAR file

Use the following task to create the deployment profile for the EAR file.

To create the deployment profile for the EAR file:

1. Access the Create Deployment Profile page by clicking the Application menu and selecting Application Properties. The Application Properties page displays.

2. From the left pane of the Application Properties page, select **Deployment**. The Deployment profiles and options display.

3. Click **New**. The Create Deployment Profile page displays.

4. Accept the Deployment Profile Name and click **OK**. The General page displays.

5. Accept the General page values and click **OK**. The Deployment profiles, including the deployment profile that you just created, and options display.

6. In the Deployment page, define the deployment profiles as follows:

    -   Select the **Auto Generate and Synchronize weblogic-jdbc.xml Descriptors During Deployment** option.

    -   Deselect the **Application Policies** and **Credentials** options.

    -   Select the **Users and Groups** option.

7. From the Deployment Profiles list, select the deployment profile that you created and click **Edit**. The Edit EAR Deployment Profile Properties page displays.

8. From the left pane, select **Application Assembly**. The Application Assembly page displays.

9. In the Java EE Modules area, locate the Model.jpr module and confirm that the items listed within this module are correct. Click **OK**. The Application Properties page displays.

10. Click **OK** to exit the Application Properties page.

4.5.2.3 Deploying the application

Use the following task to deploy the application.

To deploy the application:

-   Click the Application menu, then highlight Deploy, and then select the application to begin the deployment process.

5

Integrating Oracle Business Intelligence Presentation Services With an Oracle WebCenter Portal Application

This chapter describes how to integrate the Oracle BI Presentation Catalog with a WebCenter Portal application, and how to add the Oracle BI Composer functionality to a WebCenter Portal application or to WebCenter Portal: Spaces.

This chapter includes the following topics:

- Section 5.1, "Overview of the Oracle BI Presentation Services and WebCenter Portal Application Integration"

- Section 5.2, "Creating a WebCenter Portal Application that Integrates with Oracle BI Presentation Services"

- Section 5.3, "Can I Add BI Composer to a WebCenter Portal Application or to WebCenter Portal: Spaces?"

- Section 5.4, "Adding BI Composer to a WebCenter Portal Application or to WebCenter Portal: Spaces"

5.1 Overview of the Oracle BI Presentation Services and WebCenter Portal Application Integration

You can use JDeveloper to create WebCenter Portal applications that integrate with Oracle Business Intelligence Presentation Services. At runtime, these applications include the Oracle BI Presentation Catalog in the WebCenter Portal 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 Portal application pages. Any filters, prompts, and actions links included in the business intelligence objects work within the WebCenter Portal pages.

At runtime, WebCenter Portal users can expand and browse the Oracle BI Presentation Catalog's folders to view an analysis' views. The following view types display in the Oracle BI Presentation Catalog: table, pivot table, chart, funnel chart, gauge, narrative, ticker and title. The following view types do not display in the Oracle BI Presentation 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 Portal application pages.

For more information, see "Creating a WebCenter Portal Application that Integrates with Oracle BI Presentation Services".

5.2 Creating a WebCenter Portal Application that Integrates with Oracle BI Presentation Services

To properly create a WebCenter Portal application that integrates with Oracle BI Presentation Services, you must configure the application's library settings, update the web.xml and default-catalog.xml files, and have properly configured security.

For information describing how to configure a WebCenter Portal application to integrate with Oracle Business Intelligence Presentation Services, see "Integrating Oracle Business Intelligence Presentation Services" in *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*.

5.3 Can I Add BI Composer to a WebCenter Portal Application or to WebCenter Portal: Spaces?

BI Composer is a simple-to-use wizard that allows users to quickly and easily edit or view the design of Oracle BI EE analyses from a WebCenter Portal application or to WebCenter Portal: Spaces.

When configured, the BI Composer interface allows WebCenter Portal users to edit the analysis by selecting or modifying the analysis' columns, add views (for example, table and graphs), sort and filter data, and apply conditional formatting to data appearing in tables.

For more information about setting up this integration, see "Adding BI Composer to a WebCenter Portal Application or to WebCenter Portal: Spaces".

For information about using the BI Composer interface, see Chapter 14 "Using BI Composer to Work With Analysis" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

5.4 Adding BI Composer to a WebCenter Portal Application or to WebCenter Portal: Spaces

To add BI Composer to a WebCenter Portal application or to WebCenter Portal: Spaces, you must extend the WebCenter Portal domain to include the BI Composer implementation libraries, add library references to the WebCenter Portal application, and specify the WebCenter Portal BI Composer's viewer and editor container task flow parameters. For WebCenter Portal: Spaces, you must also add task flow permission security and apply the domain extension template to deploy the BI Composer libraries.

5.4.1 Checking for Installed JDeveloper Extensions

Oracle JDeveloper offers the following Oracle BI EE extensions:

- Business Intelligence ADF Task Flow
- Business Intelligence ADF View Regions
- Business Intelligence Composer
- Business Intelligence Logical SQL View Objects
- Business Intelligence Soap Connection

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

1. In JDeveloper, click the **Help** menu and click **About**. The "About Oracle JDeveloper 11g" dialog displays where you can confirm the release number.

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.

   Note that all Oracle BI EE extensions contain the "bi-" prefix. If no extensions begin with the "bi-" prefix, then you do not have the required extensions.

5.4.2 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 Oracle BI EE extensions. If you create a WebCenter Portal application that integrates with Oracle BI EE, then you need to also install the WebCenter Portal framework.

   For a list of the available Oracle BI EE extensions, see"Checking for Installed JDeveloper Extensions".

5.4.3 How to extend the WebCenter Portal domain

Perform this procedure to extend the WebCenter Portal domain to include the BI Composer implementation libraries. Perform this procedure for a WebCenter Portal application and for WebCenter Portal: Spaces.

For information about how to perform this procedure, see "How to Extend the Domain".

5.4.4 How to Add the BI Composer Library Reference

Use the following procedure to add the required library reference to the WebCenter Portal application. You do not need to perform this procedure for WebCenter Portal: Spaces.

1. In JDeveloper, go to the Application Navigator, open the Projects tab, navigate to the project's WEB-INF subdirectory, and open the weblogic.xml document.

2. Add the following library reference:

```
<library-ref>
  <library-name>oracle.bi.composer</library-name>
</library-ref>
```

3. Open the application's weblogic-application.xml document. To do this step, go to the Application Navigator, open the Application Resources tab, and navigate to the Descriptors subdirectory where the weblogic-application.xml document is stored.

4. Add the following library reference:

```
<library-ref>
  <library-name>oracle.bi.jbips</library-name>
</library-ref>
```

5.4.5 How to Add Task Flow Permission Security

If you are configuring BI Composer with WebCenter Portal: Spaces, use the following procedure to create a permission security entry for the BI Composer task flow. You do not need to perform this procedure for a WebCenter Portal application.

1. In Fusion Middleware Control, expand the WebCenter node.

2. Select WebCenter Spaces, right click, select **Security**, and then select **Application Policies**. The Applications Policies dialog displays.

3. In the Search pane, click the **Select Application Stripe to Search** box.

4. Click the list and select **web center** and click the **Search application security grants** button. The search results display.

5. From the search results, select the principal to which you want to add permissions. Click **Edit**. The Edit Application Grant dialog displays.

6. In the Permissions pane, select **Add**. The Add Permissions dialog displays.

7. In the Customize pane, specify the permission information. Note the following:

   - In the **Permission Class** field, enter
     oracle.adf.controller.security.TaskFlowPermission

   - In the **Resource Name** field, enter
     /WEB-INF/oracle/bi/adf/apps/answers/webcenter/flow/BIWebCenterTF.xml#BIWebCenterTF

   - In the **Permission Actions** field, enter customize, personalize, view

8. Click **OK**.

5.4.6 How to Specify the BI Composer WebCenter Portal Viewer and Editor Container Task Flow Parameters

Use the following information to initialize the taskflow with values for parameters based on other elements in the page.

The container task flow aggregates the editor and viewer task flows and allows the user to switch between the editor and viewer modes at runtime. Note that if the BI Composer libraries are not available in a WebCenter Portal application or WebCenter Portal: Spaces, then only the BI Composer's viewer is exposed and the edit functionality is disabled. However, if the BI Composer libraries are available, then the container task flow is used to expose the BI Composer's viewer and editor functionalities.

The task flow document location is:

```
WEB-INF/oracle/bi/adf/apps/answers/webcenter/flow/BIWebCenterTF.xml
```

The task flow's ID is:

BIWebCenterTF

For more information about adding parameters to a task flow, see Chapter 16: Using Parameters in Task Flows in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*

Table 5–1 lists each task flow parameter, its description, type, default value, and if it is required.

Table 5–1   BI Composer WebCenter Portal Viewer and Editor Task Flow Parameters

Name	Description	Type	Required?	Default Value
biCatalogPath	The catalog path for the analysis being viewed or edited.	String	No	null
connectionId	The ID of the deployed Oracle BI EE Presentation Services connection.	String	No	null
obipsType	The type of content being viewed or edited. Use "biReportContent" for analysis, "biDashboardContent" for dashboard, and "biScorecardContent" for scorecard.	String	No	null
subItem	The sub item name of the Oracle BI content (for example, analysis view or dashboard page). This parameter applies to the viewer, only.	String	No	null
promptfilterxml	The prompt or filter's XML. This parameter applies to the viewer, only.	String	No	null
width	The Oracle BI object's width. This parameter applies to the viewer, only.	String	No	null
height	The Oracle BI object's content height. This parameter applies to the viewer, only.	String	No	null

Table 5–1 (Cont.) BI Composer WebCenter Portal Viewer and Editor Task Flow

Name	Description	Type	Required?	Default Value
viewReportLinks	This comma separated value indicates report links to be shown for the report.	String	No	null Valid values are: ■ r Refresh. Displays the Refresh link. ■ f Print. Displays the Print link. ■ d Download. Displays the Export link. ■ - None. Displays no report links. For example, use f,r,d to show all supported links. Null is treated as the default for none.

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

Administration Server

Part of the WebLogic server domain and runs the processes that manage Oracle Business Intelligence components. The Administration Server includes the Oracle WebLogic Server Administration Console, Oracle Fusion Middleware Control, and JMX MBeans. For a Simple Install type, the Administration Server also includes Java components for Oracle Business Intelligence such as Oracle BI Publisher and Oracle Real-Time Decisions.

See also **Fusion Middleware Control**, **Java components** and **Managed Server**.

Administration Tool

See **Oracle BI Administration Tool**.

advanced trellis

A trellis view that can display multiple visualization types within its grid, for example, Spark Line graphs, Spark Bar graphs, and numbers. Each visualization type displays a different measure.

You can think of an advanced trellis as a pivot table, except that for each measure you add to the pivot table, you can optionally associate a dimension and render that dimension as a spark graph visualization.

See also **trellis** and **visualization**.

agent

Enables you to automate your business processes. You can use agents to provide event-driven alerting, scheduled content publishing, and conditional event-driven action execution.

Agents can dynamically detect information-based problems and opportunities, determine the appropriate individuals to notify, and deliver information to them through a wide range of devices (email, phones, and so on).

aggregate persistence

A feature that automates the creation and loading of aggregate tables and their corresponding Oracle Business Intelligence metadata mappings to enable aggregate navigation.

aggregate table

A table that stores precomputed results from measures that have been aggregated over a set of dimensional attributes. Each aggregate table column contains data at a given set of levels. For example, a monthly sales table might contain a precomputed sum of the revenue for each product in each store during each month. Using aggregate tables optimizes performance.

aggregation rule

In an Oracle BI repository, a rule applied to a logical column or physical cube column that specifies a particular aggregation function to be applied to the column data, such as SUM.

In Presentation Services, users can see the rules that have been applied in the repository. Users can also change the default aggregation rules for measure columns.

alias table

A physical table that references a different physical table as its source. You can use alias tables to set up multiple tables, each with different keys, names, or joins, when a single physical table must serve in different roles. Because alias table names are included in physical SQL queries, you can also use alias tables to provide meaningful table names, making the SQL statements easier to read.

analysis

A query that a user creates on the Criteria tab in Presentation Services. An analysis can optionally contain one or more filters or selection steps to restrict the results.

See also **filter** and **selection step**.

analysis criteria

Consists of the columns, filters, and selection steps that you specify for an analysis.

See also **analysis**.

analysis prompt

A prompt that is added to an analysis. When the user selects a prompt value, that value then determines the content that displays in the analysis that contains the prompt, only.

See **dashboard prompt** and **prompt**.

attribute

The details of a dimension in an Oracle BI repository. Attributes usually appear as columns of a dimension table.

attribute column

In Presentation Services, a column that holds a flat list of values that are also known as members. No hierarchical relationship exists between these members, as is the case for members of a hierarchical column. Examples include ProductID or City.

See **hierarchical column**.

BI Composer

BI Composer is a simple-to-use wizard that enables you to quickly and easily create, edit, or view analyses without the complexities of the Analysis editor.

BI domain

Contains configurable system components (the coreapplication) and Java components (the WebLogic server domain), and includes the web-based management tools and applications that use resources.

A BI domain can be a set of middleware homes spread across one or more physical servers.

See also **BI instance**.

BI instance

Refers to the system components (coreapplication) of a BI domain

See also **BI domain**.

BI object

A piece of business intelligence content that is created with Presentation Services and saved to the Oracle BI Presentation Catalog. Examples of BI objects include analyses, dashboards, dashboard pages, scorecards, and KPIs.

BI Search

A search tool that resides outside of Presentation Services. BI Search is available from the Home Page after the administrator adds a link to the BI Search URL. BI Search provides a mechanism for searching for objects in the Oracle BI Presentation Catalog that is similar to a full-text search engine.

bookmark link

Captures the path to a dashboard page and all aspects of the page state.

See **prompted link**.

bridge table

A table that enables you to resolve many-to-many relationships between two other tables.

briefing book

See **Oracle BI Briefing Books**.

business model

An object in the Oracle BI repository that contains the business model definitions and the mappings from logical to physical tables. Business models are always dimensional, unlike objects in the Physical layer, which reflect the organization of the data sources. Each business model contains logical tables, columns, and joins.

Business Model and Mapping layer

A layer of the Oracle BI repository that defines the business, or logical, model of the data and specifies the mapping between the business model and the Physical layer schemas. This layer can contain one or more business models.

The Business Model and Mapping layer determines the analytic behavior that is seen by users, and defines the superset of objects available to users. It also hides the complexity of the source data models.

business owner

The person responsible for managing and improving the business value and performance of a KPI or scorecard object, such as an objective, cause & effect map, and so on.

catalog

See **Oracle BI Presentation Catalog**.

cause & effect map

A component of a scorecard that lets you illustrate the cause and effect relationships of an objective or KPI.

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

chronological key

A column in a time dimension that identifies the chronological order of the members within a dimension level. The key must be unique at its level.

Cluster Controller

A process that serves as the first point of contact for new requests from Presentation Services and other clients. The Cluster Controller determines which Oracle BI Server in the cluster to direct the request to based on Oracle BI Server availability and load. It monitors the operation of servers in the cluster, including the Oracle BI Scheduler instances. The Cluster Controller is deployed in active-passive configuration.

column

In an Oracle BI repository, columns can be physical columns, logical columns, or presentation columns.

In Presentation Services, indicates the pieces of data that an analysis returns. Together with filters and selection steps, columns determine what analyses contain. Columns also have names that indicate the types of information that they contain, such as Account and Contact.

See also **analysis**, **attribute column**, **hierarchical column**, and **measure column**.

column filter

See **filter**.

column prompt

A type of filter that enables you to build specific value prompts on a data column to either exist alone on the dashboard or analysis or to expand or refine existing dashboard and analysis filters.

See also **prompt**.

complex join

A join in the Physical layer of an Oracle BI repository that uses an expression other than equals.

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 uses the Action Framework to pass content from the business intelligence object to another region on an ADF page.

See also **action**, **Action Framework**, and **action link**.

criteria

See **analysis criteria**.

cube

An OLAP (online analytical processing) data structure that lets data be analyzed more quickly and with greater flexibility than structures in relational databases. Cubes are made up of measures and organized by dimensions. Cubes in multidimensional data sources roughly correspond to star schemas in relational database models.

currency prompt

A prompt that enables the user to change the currency type that displays in the currency columns on an analysis or dashboard.

See also **prompt**.

custom view

A component of a scorecard that lets you show a customized view of your business and strategy data.

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

dashboard

An object that provides personalized views of corporate and external information. A dashboard consists of one or more pages. Pages can display anything that you can access or open with a web browser, such as results of analyses, images, alerts from agents, and so on.

dashboard prompt

A prompt that is added to the dashboard. When the user selects a prompt value, that value then determines the content that displays in all analyses that are included on the dashboard.

See **analysis prompt** and **prompt**.

Dashboard URL

Used for incorporating or referencing the content of a specific dashboard in external portals or applications. It has several forms and optional arguments that you can use to control its behavior.

data source name (DSN)

A data structure that contains the information about a specific database, typically used by an ODBC driver to connect to the database. The DSN contains information such as the name, directory, and driver of the database.

Connection pool objects in the Physical layer of the Oracle BI repository contain DSN information for individual data sources.

database hint

Instructions placed within a SQL statement that tell the database query optimizer the most efficient way to execute the statement. Hints override the optimizer's execution plan, so you can use hints to improve performance by forcing the optimizer to use a more efficient plan. Hints are supported only for Oracle Database data sources.

dimension

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

A dimension might contain one or more (unnamed) hierarchies. There are two types of logical dimensions: dimensions with level-based hierarchies (structure hierarchies), and dimensions with parent-child hierarchies (value hierarchies).

A particular type of level-based dimension, called a time dimension, provides special functionality for modeling time series data.

See also **hierarchy**.

dimension table

A logical table that contains columns used by a particular dimension. A dimension table cannot be a fact table.

See also **fact table**.

driving table

A mechanism used to optimize the manner in which the Oracle BI Server processes multi-database joins when one table is very small (the driving table) and the other table is very large.

DSN

See **data source name (DSN)**.

event polling table

Event polling tables (also called event tables) provide information to the Oracle BI Server about which physical tables have been updated. They are used to keep the query cache up-to-date. The Oracle BI Server cache system polls the event table, extracts the physical table information from the rows, and purges stale cache entries that reference those physical tables.

Essbase

A multidimensional database management system available from Oracle that provides a multidimensional database platform upon which to build business intelligence applications. Also referred to as Oracle's Hyperion Essbase.

fact table

In an Oracle BI repository, a logical table in the Business Model and Mapping layer that contains measures and has complex join relationships with dimension tables.

See also **dimension table**.

filter

Criteria that are applied to attribute and measure columns to limit the results that are displayed when an analysis is run. For measure columns, filters are applied before the query is aggregated and affect the query and thus the resulting values.

See also **prompt** and **selection step**.

foreign key

A column or a set of columns in one table that references the primary key columns in another table.

fragmentation content

The portion, or fragment, of the set of data specified in a logical table source when the logical table source does not contain the entire set of data at a given level. Fragmentation content is defined by the logical columns that are entered in the **Fragmentation content** box in the Content tab of the Logical Table Source dialog box.

Fusion Middleware Control

Provides web-based management tools that enable you to monitor and configure Fusion Middleware components.

global header

An object in the user interface for Oracle BI Presentation Services that contains links and options that enable the user to quickly begin a task or locate a specific object within the Oracle BI Presentation Catalog. The global header always displays in the Presentation Services user interface, thus enabling users to quickly access links and search the catalog without having to navigate to the Home Page or Catalog page.

Go URL

Used to incorporate specific business intelligence results into external portals or applications. The Go URL is used when you add a result to your favorites or add a link to a request to a dashboard or external web site. It has several forms and optional arguments that you can use to control its behavior.

hierarchical column

In Presentation Services, a column that holds data values that are organized using both named levels and parent-child relationships. This column is displayed using a tree-like structure. Individual members are shown in an outline manner, with lower-level members rolling into higher-level members. For example, a specific day belongs to a particular month, which in turn is within a particular year. Examples include Time or Geography.

See also **attribute column**.

hierarchy

In an Oracle BI repository, a system of levels in a logical dimension that are related to each other by one-to-many relationships. All hierarchies must have a common leaf level and a common root (all) level.

Hierarchies are not modeled as separate objects in the metadata. Instead, they are an implicit part of dimension objects.

See also **dimension**, **logical level**, and **presentation hierarchy**.

hierarchy level

In Presentation Services, an object within a hierarchical column that either rolls up or is rolled up from other levels. Corresponds to a presentation level in an Oracle BI repository.

See also **presentation level**.

home page

Provides an intuitive, task-based entry way into the functionality of Presentation Services. The Home page is divided into sections that enable you to quickly begin specific tasks, locate an object, or access technical documentation.

image prompt

A prompt that provides an image with different areas mapped to specific values. The user clicks an image area to select the prompt value that populates the analysis or dashboard.

See also **prompt**.

initialization block

Used to initialize dynamic repository variables, system session variables, and nonsystem session variables. An initialization block contains the SQL statements that are executed to initialize or refresh the variables that are associated with that block.

initiative

Used in a scorecard, an initiative is a time-specific task or project that is necessary to achieve objectives. As such, you can use initiatives that support objectives as milestones as they reflect progress toward strategy targets.

See also **objective** and **Oracle Scorecard and Strategy Management**.

inner graph

A nested graph, inside the grid of a trellis graph. Each inner graph has its own dimensionality as specified in the Visualization area of the Layout pane.

See also **trellis** and **outer edge**.

Java components

Fusion Middleware Control components that are deployed as one or more Java EE applications (and a set of resources) and are managed by Node Manager.

See also **Node Manager**.

key performance indicator (KPI)

A measurement that defines and tracks specific business goals and strategic objectives. KPIs often times roll up into larger organizational strategies that require monitoring, improvement, and evaluation. KPIs have measurable values that usually vary with time, have targets to determine a score and performance status, include dimensions to allow for more precise analysis, and can be compared over time for trending purposes and to identify performance patterns.

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

KPI watchlist

A method of distributing KPIs to end users. A watchlist is a collection of KPIs that are built by adding the KPIs that are stored in the Oracle BI Presentation Catalog. After a KPI watchlist is built and saved, it is stored as a catalog object and can be added to dashboards and scorecards.

See also **key performance indicator (KPI)**, **watchlist**, and **Oracle Scorecard and Strategy Management**.

level

See **hierarchy level**.

logical display folder

Folders used to organize objects in the Business Model and Mapping layer of an Oracle BI repository. They have no metadata meaning.

logical join

Joins that express relationships between logical tables. Logical joins are *conceptual*, rather than physical, joins. In other words, they do not join to particular keys or columns. A single logical join can correspond to many possible physical joins.

logical layer

See **Business Model and Mapping layer**.

logical level

In an Oracle BI repository, a component of a level-based hierarchy that either rolls up or is rolled up from other levels.

Parent-child hierarchies have implicit, inter-member levels between ancestors and descendants that are not exposed as logical level objects in the metadata. Although parent-child hierarchies also contain logical level objects, these levels are system generated and exist to enable aggregation across all members only.

See also **dimension** and **hierarchy**.

Logical SQL

The SQL statements that are understood by the Oracle BI Server. The Oracle BI Server Logical SQL includes standard SQL, plus special functions (SQL extensions) like AGO, TODATE, EVALUATE, and others.

Clients like Presentation Services send Logical SQL to the Oracle BI Server when a user makes a request. In addition, Logical SQL is used in the Business Model and Mapping layer to enable heterogeneous database access and portability. The Oracle BI Server transforms Logical SQL into physical SQL that can be understood by source databases.

logical table

A table object in the Business Model and Mapping layer of an Oracle BI repository. A single logical table can map to one or more physical tables. Logical tables can be either fact tables or dimension tables.

See also **dimension table** and **fact table**.

logical table source

Objects in the Business Model and Mapping layer of an Oracle BI repository that define the mappings from a single logical table to one or more physical tables. The physical to logical mapping can also be used to specify transformations that occur between the Physical layer and the Business Model and Mapping layer, and to enable aggregate navigation and fragmentation.

Managed Server

An individual J2EE application container (JMX MBean container). It provides local management functions on individual hosts for Java components and system components contained within the local middleware home, and refers to the Administration Server for all of its configuration and deployment information.

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

MDS

Oracle Metadata Services. A core technology of the Application Development Framework. MDS provides a unified architecture for defining and using metadata in an extensible and customizable manner.

See also **MDS XML**.

MDS XML

An XML format that is compatible with Oracle Metadata Services. MDS XML is a supported format for the Oracle BI repository. It enables integration with third-party source control management systems for offline repository development.

MDS XML format is different from the XML format generated by the Oracle BI Server XML API.

See also **MDS**, **Oracle BI repository**, and **Oracle BI Server XML API**.

measure column

A column that can change for each record and can be added up or aggregated. Typical measures are sales dollars and quantity ordered. Measures are calculated from data sources at query time.

Measure columns are displayed in the Oracle BI repository, usually in fact tables, or in Presentation Services.

metadata

Data about data. Metadata objects include the descriptions of schemas (such as tables, columns, data types, primary keys, foreign keys, and so on) and logical constructs (like fact tables, dimensions, and logical table source mappings).

The Oracle BI repository is made up of the metadata used by the Oracle BI Server to process queries.

metadata dictionary

A static set of XML documents that describe metadata objects, such as a column, including its properties and relationships with other metadata objects. A metadata dictionary can help users obtain more information about metrics or attributes for repository objects.

microchart

A tiny graph displayed in a grid along with other tiny graphs and numbers, comprising the data cell contents of an advanced trellis view. In Oracle BI EE, a microchart is always a spark graph.

See also **advanced trellis** and **spark graph**.

mission statement

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

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

multi-database join

A join between two tables in an Oracle BI repository, where each table resides in a different database.

Node Manager

A daemon process that provides remote server start, stop, and restart capabilities when Java processes become unresponsive or terminate unexpectedly.

See also **Java components**.

OCI

See **Oracle Call Interface (OCI)**.

ODBC

See **Open Database Connectivity (ODBC)**.

object properties

Information about an object and attributes that the owner can assign to an object. Examples of properties include name, description, date stamps, read-only access, and do not index flag.

See also **permissions**.

objective

A required or desired outcome in a scorecard that forms your corporate strategy.

See also **initiative** and **Oracle Scorecard and Strategy Management**.

offline mode

In the Oracle BI Administration Tool, a mode where a repository builder can edit a repository that is not loaded into the Oracle BI Server.

online mode

In the Oracle BI Administration Tool, a mode where a repository builder can edit a repository while it is available for query operations. Online mode also allows user session monitoring for users connected to the subject areas in the repository.

opaque view

A Physical layer table that consists of a `SELECT` statement. In the Oracle BI repository, opaque views appear as view tables in the physical databases, but the view does not actually exist.

Open Database Connectivity (ODBC)

A standard interface used to access data in both relational and 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, the Business Model and Mapping layer, and the Presentation layer.

Oracle BI Briefing Books

A collection of static or updatable snapshots of dashboard pages, individual analyses, and BI Publisher reports. You can download briefing books in PDF or MHTML format for printing and viewing. You also can update, schedule, and deliver briefing books using agents.

Oracle BI JavaHost

A service that gives Presentation Services the ability to use functionality that is provided in Java libraries to support components such as graphs. The services are provided based on a request-response model.

Oracle BI Logical SQL View Object

Provides the developer the ability to create a Logical SQL statement to access the Oracle BI Server and fetch business intelligence data and bind it to native ADF components for inclusion on an ADF page. This view object uses a BI JDBC connection to the Oracle BI Server.

Oracle BI Presentation Catalog

Stores business intelligence objects, such as analyses and dashboards, and provides an interface where users create, access, and manage objects, and perform specific object-based tasks (for example, export, print, and edit). The catalog is organized into folders that are either shared or personal.

Oracle BI Presentation Services

Provides the framework and interface for the presentation of business intelligence data to web clients. It maintains a Presentation Catalog service on the file system for the customization of this presentation framework. It is a standalone process and communicates with the Oracle BI Server using ODBC over TCP/IP. It consists of components that are known as Answers, Delivers, and Interactive Dashboards.

See also **ODBC**; **Oracle BI Server**; **Oracle BI Presentation Catalog**; **Oracle BI Presentation Services 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

The set of Oracle Business Intelligence metadata that defines logical schemas, physical schemas, physical-to-logical mappings, aggregate table navigation, and other constructs. Oracle BI repositories can be in binary (RPD) format, in which repository metadata is contained in a single file with an extension of .rpd, or in a set of MDS XML documents. MDS XML format repositories are used for offline development only and cannot be loaded into the Oracle BI Server. Oracle BI repositories in both formats can be edited using the Oracle BI Administration Tool.

See also **metadata** and **Oracle BI Administration Tool**.

Oracle BI Scheduler

An extensible scheduling application for scheduling results to be delivered to users at specified times. It is the engine behind the Oracle BI Delivers feature.

See also **results**.

Oracle BI Server

A standalone process that maintains the logical data model that it provides to Presentation Services and other clients through ODBC. Metadata is maintained for the data model in a local proprietary file called the repository file. The Oracle BI Server processes user requests and queries underlying data sources.

Oracle BI Server XML API

Provides utilities to create a generic, XML-based representation of the Oracle BI repository metadata. You can use this XML file version of the repository to programmatically modify the metadata. The Oracle BI Server XML API objects correspond to metadata repository objects in an RPD file. These objects differ from XML objects in the Oracle BI Presentation Catalog.

The XML generated by the Oracle BI Server XML API is different from the MDS XML format used for Oracle BI repositories integrated with third-party source control management systems.

See also **MDS XML**.

Oracle Business Intelligence Mobile

Oracle Business Intelligence Mobile allows you to view Oracle BI EE content on supported mobile devices such as the Apple iPhone and Apple iPad.

Using Oracle Business Intelligence Mobile, you can view and analyze BI content such as analyses and dashboards, BI Publisher content, scorecard content, and content delivered by agents.

Oracle Business Intelligence Web Services

See Oracle Business Intelligence Session-Based Web Services and Oracle Business Intelligence Web Services for SOA.

Oracle Business Intelligence Session-Based Web Services

An API that implements SOAP. These web services are designed for programmatic use, where a developer uses one web service to invoke many different business intelligence objects. These web services provide functionality on a wide range of Presentation Services operations. These web services enable the developer to extract results from Oracle BI Presentation Services and deliver them to external applications, perform Presentation Services management functions, and execute Oracle Business Intelligence alerts (known as Intelligent Agents).

See also Oracle Business Intelligence Web Services for SOA.

Oracle Business Intelligence Web Services for SOA

Contains three web services, ExecuteAgent, ExecuteAnalysis, and ExecuteCondition, which are hosted by the bimiddleware J2EE application. These web services are designed to enable developers to use third-party web services clients (for example, Oracle SOA Suite) to browse for and include business intelligence objects in service oriented architecture components.

See also Oracle Business Intelligence Session-Based Web Services.

Oracle Call Interface (OCI)

A connection interface that the Oracle BI Server can use to connect to Oracle Database data sources. You should always use OCI when importing metadata from or connecting to an Oracle Database.

Oracle OLAP

Oracle Database has an OLAP Option that provides an embedded, full-featured online analytical processing server.

Oracle Business Intelligence supports Oracle OLAP as a data source. When you import metadata from an Oracle OLAP source, the Oracle OLAP objects appear in the Physical layer of the Administration Tool. Oracle OLAP objects include Analytic Workspaces, which are containers for storing related cubes.

Oracle Process Manager and Notification Server (OPMN)

A process management tool that manages all system components (server processes), and supports both local and distributed process management, automatic process recycling and the communication of process state (up, down, starting, stopping). OPMN detects process unavailability and automatically restarts processes).

See also **system components**.

Oracle Scorecard and Strategy Management

A performance management tool that lets you describe and communicate your business strategy. You can drive and assess your corporate strategy and performance from the top of your organization down, or from the bottom up.

Oracle Technology Network (OTN)

A repository of technical information about Oracle's products where you can search for articles, participate in discussions, ask the user community technical questions, and search for and download Oracle products and documentation.

outer edge

The outer edges are the parts of a trellis view that border the inner graphs. These include the column and row headers, the section headers, and so on.

See also **trellis** and **inner graph**.

parent-child hierarchy

A hierarchy of members that all have the same type. All the dimension members of a parent-child hierarchy occur in a single data source. In a parent-child hierarchy, the inter-member relationships are parent-child relationships between dimension members.

See also **dimension**.

parent-child relationship table

A table with values that explicitly define the inter-member relationships in a parent-child hierarchy. Also called a closure table.

pass-through calculation

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

permissions

Specify which users can access an object, and limit how users can interact with an object. Examples of permissions include write, delete, and change permissions.

See **object properties**.

perspective

A category in your organization with which to associate initiatives, objectives, and KPIs in a scorecard. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

See also **initiative**, **key performance indicator (KPI)**, **objective**, and **Oracle Scorecard and Strategy Management**.

physical catalog

An object in the Physical layer of an Oracle BI repository that groups different schemas. A catalog contains all the schemas (metadata) for a database object.

physical display folder

Folders that organize objects in the Physical layer of an Oracle BI repository. They have no metadata meaning.

physical join

Joins between tables in the Physical layer of an Oracle BI repository.

Physical layer

A layer of the Oracle BI repository that contains objects that represent physical data constructs from back-end data sources. The Physical layer defines the objects and relationships available for writing physical queries. This layer encapsulates source dependencies to enable portability and federation.

physical schema

An object in the Physical layer of an Oracle BI repository that represents a schema from a back-end database.

physical table

An object in the Physical layer of an Oracle BI repository, usually corresponding to a table that exists in a physical database.

See also **Physical layer**.

presentation hierarchy

An object in the Presentation layer of an Oracle BI repository that provides an explicit way to expose the multidimensional model in Presentation Services and other clients. Presentation hierarchies expose analytic functionality such as member selection, custom member groups, and asymmetric queries. Users can create hierarchy-based queries using presentation hierarchies.

In Presentation Services, presentation hierarchies are displayed as hierarchical columns.

See also **hierarchical column** and **presentation level**.

Presentation layer

Provides a way to present customized, secure, role-based views of a business model to users. It adds a level of abstraction over the Business Model and Mapping layer in the Oracle BI repository. The Presentation layer provides the view of the data seen by users who build analyses in Presentation Services and other client tools and applications.

See also **Business Model and Mapping layer**.

presentation level

In the Oracle BI repository, a component of a presentation hierarchy that either rolls up or is rolled up from other levels. Presentation levels are displayed as levels within hierarchical columns in Presentation Services.

See also **hierarchy level** and **presentation hierarchy**.

Presentation Services

See **Oracle BI Presentation Services**.

Presentation Services server

See **Oracle BI Presentation Services server**.

presentation table

An object in the Presentation layer of an Oracle BI repository that is used to organize columns into categories that make sense to the user community. A presentation table

can contain columns from one or more logical tables. The names and object properties of the presentation tables are independent of the logical table properties.

primary key

A column (or set of columns) where each value is unique and identifies a single row of a table.

process instance

A unique process on an individual workstation that is associated with a BI instance.

See also **BI instance**.

prompt

A type of filter that enables the content designer to build and specify data values or the end user to choose specific data values to provide a result sets for an individual analysis or multiple analyses included on a dashboard or dashboard page. A prompt expands or refines existing dashboard and analysis filters.

The types of prompts are column prompts, currency prompts, image prompts, and variable prompts.

See also **column prompt**, **currency prompt**, **filter**, **image prompt**, and **variable prompt**.

prompted link

Captures the path to a dashboard page and a simplified presentation of the dashboard prompt.

See **bookmark link**.

query

Contains the underlying SQL statements that are issued to the Oracle BI Server. You do not have to know a query language to use Oracle Business Intelligence.

query cache

A facility to store query results for use by other queries.

ragged hierarchy

See **unbalanced hierarchy**.

report

The response returned to the user from the execution of a query created using Oracle BI Publisher. Reports can be formatted, presented on a dashboard page, saved in the Oracle BI Presentation Catalog, and shared with other users.

See also **analysis**.

repository

See **Oracle BI repository**.

repository variable

See **variable**.

results

The output returned from the Oracle BI Server for an analysis.

See also **analysis**.

scorecard

See **Oracle Scorecard and Strategy Management**.

selection step

A choice of values that is applied after the query is aggregated that affects only the members displayed, not the resulting aggregate values. Along with filters, selection steps restrict the results for an analysis.

See also **analysis** and **filter**.

session variable

See **variable**.

simple trellis

A trellis view that displays inner visualizations that are all the same type, such as all scatter graphs. The inner visualizations all use a common axis, also known as a synchronized scale.

See also **trellis**, **synchronized scale**, and **visualization**.

skip-level hierarchy

A hierarchy where some members do not have a value for a particular ancestor level. For example, in the United States, the city of Washington in the District of Columbia does not belong to a state. The expectation is that users can still navigate from the country level (United States) to Washington and below without the need for a state.

See also **hierarchy**.

smart watchlist

A smart watchlist is a view into a particular scorecard based on criteria that you specify. For example, a smart watchlist might show the top ten KPIs in a scorecard based on best performance or all the objectives, initiatives, and KPIs in a scorecard that are owned by a specific business owner.

See also **watchlist** and **Oracle Scorecard and Strategy Management**.

snowflake schema

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

spark graph

An embedded mini-graph that, in conjunction with other mini-graphs and numbers, illustrates a single trend. Spark graphs are also known as *sparks*.

Sparks do not include axes or labels; they get their context from the content that surrounds them. Each type of spark graph has only one measure, which is hidden; the scale is relative to itself only.

A spark graph can be of the graph subtype Spark Line, Spark Bar, or Spark Area.

See also **microchart**.

SQL

See **structured query language (SQL)**.

star schema

A relational schema that allows dimensional analysis of historical information. Star schemas have one-to-many relationships between the logical dimension tables and the logical fact table. Each star consists of a single fact table joined to a set of denormalized dimension tables.

strategy map

A component of a scorecard that shows how the objectives that have been defined for a scorecard and the KPIs that measure their progress are aligned by perspectives. It also shows cause and effect relationships.

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

strategy tree

A component of a scorecard that shows an objective and its supporting child objectives and KPIs hierarchically in a tree diagram.

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

structured query language (SQL)

A standard programming language for querying and modifying data.

See also **Logical SQL**.

subject area

In an Oracle BI repository, an object in the Presentation layer that organizes and presents data about a business model. It is the highest-level object in the Presentation layer and represents the view of the data that users see in Presentation Services. Oracle BI repository subject areas contain presentation tables, presentation columns, and presentation hierarchies.

In Presentation Services, subject areas contain folders, measure columns, attribute columns, hierarchical columns, and levels.

synchronized scale

(Applicable to simple trellis only) A synchronized scale means that all the visualizations within the trellis are viewed on the same scale, that is, they share a common axis. Having a common axis makes all graph markers easy to compare across rows and columns.

See also **simple trellis** and **visualization**.

system components

Server processes (not Java applications) that are managed by the Oracle Process Manager and Notification server (OPMN).

See also **Oracle Process Manager and Notification Server (OPMN)**.

transformation

Work that is performed on data when moving from a database to another location (sometimes another database). Some transformations are typically performed on data when it is moved from a transaction system to a data warehouse system.

trellis

Displays multidimensional data shown as a set of cells in a grid, where each cell represents a subset of data using a particular graph type. Data can be represented with graphs, microcharts, and numbers.

The trellis view has two subtypes: simple trellis and advanced trellis.

See also **advanced trellis** and **simple trellis**.

unbalanced hierarchy

A hierarchy where the leaves do not have the same depth. For example, an organization might choose to have data for the current month at the day level, data for the previous year at the month level, and data for the previous five years at the quarter level.

See also **hierarchy**.

value hierarchy

See **parent-child hierarchy**.

variable

Objects in an Oracle BI repository that are used to streamline administrative tasks and dynamically modify metadata content to adjust to a changing data environment.

Variables are of the following types:

- Repository variables have a single value at any point in time. There are two types of repository variables: static and dynamic.

- Session variables are created and assigned a value when each user logs on. There are two types of session variables: system and nonsystem.

variable prompt

Enables the user to select a value specified in the variable prompt to display on the dashboard. A variable prompt is not dependent upon column data, but enables you to manipulate, for example add or multiply, the column data on an analysis.

See also **prompt**.

virtual physical table

A physical table that is made from a stored procedure or a SELECT statement. Creating virtual tables can provide the Oracle BI Server and the underlying databases with the proper metadata to perform some advanced query requests.

vision statement

A short statement in a scorecard that describes what your organization wants to become sometime in the future. For example, it might be to become the most successful business in the South America Polypropylene Market.

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

visualization

In the context of Oracle BI EE, a visualization is the choice of graph that appears within a data cell in a trellis view. There are many visualizations from which to choose when creating a trellis view, including bar graphs, scatter graphs, and spark graphs.

See also **trellis**.

watchlist

A table that lists scorecard objects (that is, initiatives, objectives, and KPIs) that are related to a particular aspect of a scorecard or that are grouped together for a particular purpose. There are different types for watchlists for example, KPI watchlists or smart watchlists.

See also **KPI watchlist**, **smart watchlist**, and **Oracle Scorecard and Strategy Management**.

WebLogic server domain

Contains Java components that are configured to participate in the servicing of SOAP, HTTP, and other forms of requests.

WebLogic Scripting Tool (WLST)

A command-line scripting interface that enables you to configure, manage, and persist changes to WebLogic Server instances and domains and to monitor and manage server runtime events.

XML API

See **Oracle BI Server XML API**.

Index