Oracle® Fusion Middleware

System Administrator's Guide for Oracle Business Intelligence Enterprise Edition

11*g* Release 1 (11.1.1)

E10541-01

July 2010

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

E10541-01

Contents

3 Configuring the Oracle Business Intelligence System

Part II Managing Processes and Components

4 Starting and Stopping Oracle Business Intelligence

7 Managing Performance Tuning and Query Caching

Part IV Resolving Issues

8 Diagnosing and Resolving Issues in Oracle Business Intelligence

9 Managing Usage Tracking

Part V Configuring Oracle Business Intelligence

10 Configuring Repositories

11 Configuring Connections to External Systems

12 Configuring Presentation Setting Defaults

13 Configuring Mapping and Spatial Information

14 Configuring Oracle Scorecard and Strategy Management

15 Configuring Time Zones

16 Localizing Oracle Business Intelligence

17 Configuring Currency Options

18 Configuring and Managing the Oracle BI Presentation Catalog

Part VI Advanced Configuration Settings

19 Configuring and Managing Analyses and Dashboards

##

##

##

23 Configuring Resource Availability and URL Generation

Part VII Configuring Oracle BI Scheduler

24 Introducing Oracle BI Scheduler

25 Configuration Tasks for Oracle BI Scheduler

26 Using Oracle BI Scheduler Job Manager

27 Programming BI Scheduler VBScript and JScript Jobs

28 Programming BI Scheduler Java Jobs

29 Oracle BI Scheduler Job Manager Menus

Part VIII Automating Management of Oracle Business Intelligence

30 Introducing the Oracle BI Systems Management API

Part IX Managing the Life Cycle

31 Patching Oracle Business Intelligence Systems

32 Moving to a New Environment

33 Moving From a Test to a Production Environment

34 Backup and Recovery of Oracle Business Intelligence Systems

Part X Reference Information

A NQSConfig.INI File Configuration Settings

B Exchanging Metadata with Databases to Enhance Query Performance

C Managing Security for Dashboards and Analyses

D Setting Up the SA System Subject Area

E Advanced Configuration Reference

F Propagating Labels and Tooltips from Oracle ADF Data Sources

G Mapping User Interface Labels with Configuration File Elements

H Manually Updating Fusion Middleware Control Settings

Glossary

Index

Preface

Oracle Business Intelligence Enterprise Edition is a comprehensive set of enterprise business intelligence tools and infrastructure, including a scalable and efficient query and analysis server, an ad-hoc query and analysis tool, interactive dashboards, proactive intelligence and alerts, real-time predictive intelligence, and an enterprise reporting engine. Oracle Business Intelligence Enterprise Edition is designed to bring greater business visibility and insight to a wide variety of users.

The components of Oracle Business Intelligence Enterprise Edition 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 Business Intelligence Enterprise Edition provides scalability and performance with data-source specific optimized analysis generation, optimized data access, advanced calculation, intelligent caching services, and clustering.

Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition is part of the documentation set for Oracle Business Intelligence Enterprise Edition. This guide contains information about system administration tasks and includes topics on starting and stopping processes, managing logging and usage tracking, managing query caching and performance, managing scalability and high availability, and setting configuration options.

Audience

This document is intended for system administrators who are responsible for managing Oracle Business Intelligence processes, logging, caching, monitoring, high availability, and configuration.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Access to Oracle Support

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

Related Documents

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

- *Oracle Fusion Middleware Release Notes* for your platform
- *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*
- *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence*

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 System Administrators

This section describes changes to system administration features for Oracle Business Intelligence Enterprise Edition 11*g* Release 1 (11.1.1). If you are upgrading to Oracle BI EE from a previous release, then read the following information carefully, because there are significant differences in features, tools, and procedures.

This preface contains the following topics:

- New Features
- Upgrade Considerations

New Features

New system administration features in Oracle BI EE include:

- Integrated Management Experience
- Centralized System Administration
- Programmatic and Scripting Capabilities for Administrative Operations
- New Configuration Settings for Views
- Management of Users and Groups
- Enhancements for Localization

Integrated Management Experience

This release introduces a fully integrated management experience for the administration of Java and non-Java components using Fusion Middleware Control and Oracle WebLogic Server Administration Console. Because of this new integrated management experience, you no longer need to manually change configuration files for most administration tasks.

Centralized System Administration

In this release, you can now perform most system administration tasks using centralized management capabilities, including the following:

- Start and stop Oracle Business Intelligence components. See Chapter 4, "Starting and Stopping Oracle Business Intelligence" for more information.
- Increase the capacity of your system by scaling Oracle Business Intelligence components. See Chapter 5, "Scaling Your Deployment" for more information.

- Configure and monitor system availability. See Chapter 6, "Deploying Oracle Business Intelligence for High Availability" for more information.
- View performance metrics and tune the system. See Chapter 7, "Managing Performance Tuning and Query Caching" for more information.
- Diagnose and troubleshoot issues across all computers in your cluster. See Chapter 8, "Diagnosing and Resolving Issues in Oracle Business Intelligence" for more information.
- Change configuration and system preferences. See Chapter 10, "Configuring Repositories" for more information.

For information, see Chapter 2, "Managing Oracle Business Intelligence."

Programmatic and Scripting Capabilities for Administrative Operations

This release enables the automation of configuration and system management. The new BI Systems Management API Java programming interface includes a rich set of standards-based JMX MBeans to enable developers to automate administrative operations using Java and scripting technologies such as WLST (WebLogic Scripting Tool) and JPython. For information, see Chapter 30, "Introducing the Oracle BI Systems Management API."

New Configuration Settings for Views

This release provides new configuration settings for table, pivot table, graph, and gauge views. The release also introduces a new type called map view. For information on configuring these views, see Chapter 19, "Configuring and Managing Analyses and Dashboards."

Management of Users and Groups

This release introduces several enhancements and changes in how users and groups are managed. This release also uses application roles for organizing users. The Presentation Services groups from the previous release are now known as Catalog groups. For information, see Appendix C, "Managing Security for Dashboards and Analyses."

Enhancements for Localization

This release introduces several enhancements for localizing your system, including lookup tables and alias tables. For information, see Chapter 16, "Localizing Oracle Business Intelligence."

Upgrade Considerations

The following list identifies what you must be aware of if your site is upgrading from Oracle Business Intelligence 10*g* to Oracle Business Intelligence 11*g* Release 1 (11.1.1):

- The Everyone group has been replaced with the AuthenticatedUser role. For information, see Appendix C, "Managing Security for Dashboards and Analyses."

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

Part I

Administering Oracle Business Intelligence

This part explains how to administer Oracle Business Intelligence. It contains the following chapters:

- Chapter 1, "Introduction to Oracle Business Intelligence System Administration"
- Chapter 2, "Managing Oracle Business Intelligence"
- Chapter 3, "Configuring the Oracle Business Intelligence System"

1

Introduction to Oracle Business Intelligence System Administration

This chapter introduces system administration in Oracle Business Intelligence, explains what a system administrator does; describes where to get started with typical system administration tasks; describes the BI architecture; lists the tools that can help you complete system administration tasks, and provides links to system requirements and certification information.

This chapter contains the following topics:

- Section 1.1, "What are the Oracle Business Intelligence System Administration Tasks?"
- Section 1.2, "Getting Started with Managing Oracle Business Intelligence"
- Section 1.3, "What is the Oracle Business Intelligence System Logical Architecture?"
- Section 1.4, "About Oracle BI Java Components and System Components"
- Section 1.5, "What is the Administration Server?"
- Section 1.6, "What System Administration Tools are Available for Managing Oracle Business Intelligence?"
- Section 1.7, "Topics of Interest in Other Guides"
- Section 1.8, "System Requirements and Certification"

1.1 What are the Oracle Business Intelligence System Administration Tasks?

Administering an Oracle Business Intelligence system involves the following tasks:

- **Configuring a system for deployment after installation**

  Configuring metadata and content, general preferences, and default system settings.

- **Starting and stopping the system when required**

  Bringing the system up and down during system maintenance tasks.

- **Configuring security**

  Securing access to the Oracle Business Intelligence system, metadata, and data, configuring Secure Socket Layer (SSL) and Single Sign On (SSO), and integration with identity management systems.

- **Scaling out and configuring for high availability**

  Configuring the Oracle Business Intelligence system for linear scale-out and identifying and removing single points of failure.

- **Managing performance and availability**

  Monitoring service levels and tuning performance.

- **Managing and resolving issues**

  Diagnosing errors and establishing resolutions.

- **Moving a system from test to production**

  Managing the steps for moving from a test to a production environment.

- **Backing up and recovering data**

  Preparing for and recovering from unexpected events.

For more information about these tasks, see Section 1.2, "Getting Started with Managing Oracle Business Intelligence."

1.2 Getting Started with Managing Oracle Business Intelligence

Use this section to identify a task to complete, then click the corresponding link to display the appropriate content.

Table 1–1 describes the typical system administration tasks that you perform in Oracle Business Intelligence and indicates where to find related information.

Table 1–1 Oracle Business Intelligence System Administration Tasks

System Administration Task	More Information
Learning about Oracle Business Intelligence system administration	For more information, see the topics in this section. Contains information about the system architecture, components, tools, links to other related topics, and certification information.
Viewing Oracle Business Intelligence status	Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components" Also contains information about using Fusion Middleware Control and using WebLogic Server Administration Console.
Configuring Oracle Business Intelligence	Chapter 3, "Configuring the Oracle Business Intelligence System" Contains information about the available methods for updating configuration settings and where configuration files are located.
Starting and stopping Oracle Business Intelligence	Chapter 4, "Starting and Stopping Oracle Business Intelligence" Contains various topics on starting and stopping components, in addition to troubleshooting information.
Managing availability and capacity	Part III, "Scaling and Deploying for High Availability and Performance" Contains chapters about extending and scaling the environment, deploying for high availability, performance tuning, and query caching.

Table 1–1 (Cont.) Oracle Business Intelligence System Administration Tasks

System Administration Task	More Information
Diagnosing problems and resolving issues	Part IV, "Resolving Issues" Contains chapters about diagnosing and resolving issues and about usage tracking.
Configuring Oracle Business Intelligence	Part V, "Configuring Oracle Business Intelligence" Contains chapters about required configuration such as configuring repositories and connections to external systems.
Modifying advanced configuration settings	Part VI, "Advanced Configuration Settings" Contains chapters about advanced and optional configuration settings for features such as analyses, dashboards, and maps.
Configuring Oracle BI Scheduler	Part VII, "Configuring Oracle BI Scheduler"
Automating management of an Oracle Business Intelligence system	Part VIII, "Automating Management of Oracle Business Intelligence" Describes the Oracle BI Systems Management API.
Managing the life cycle.	Part IX, "Managing the Life Cycle" Contains chapters about life cycle management tasks such as patching, moving between environments, and backup and recovery.
Securing the system	■ Define administrative role membership Appendix C, "Managing Security for Dashboards and Analyses" ■ Secure middle-tier communications Secure Socket Layer (SSL) and Single Sign On (SSO) are not described in this guide. For information, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

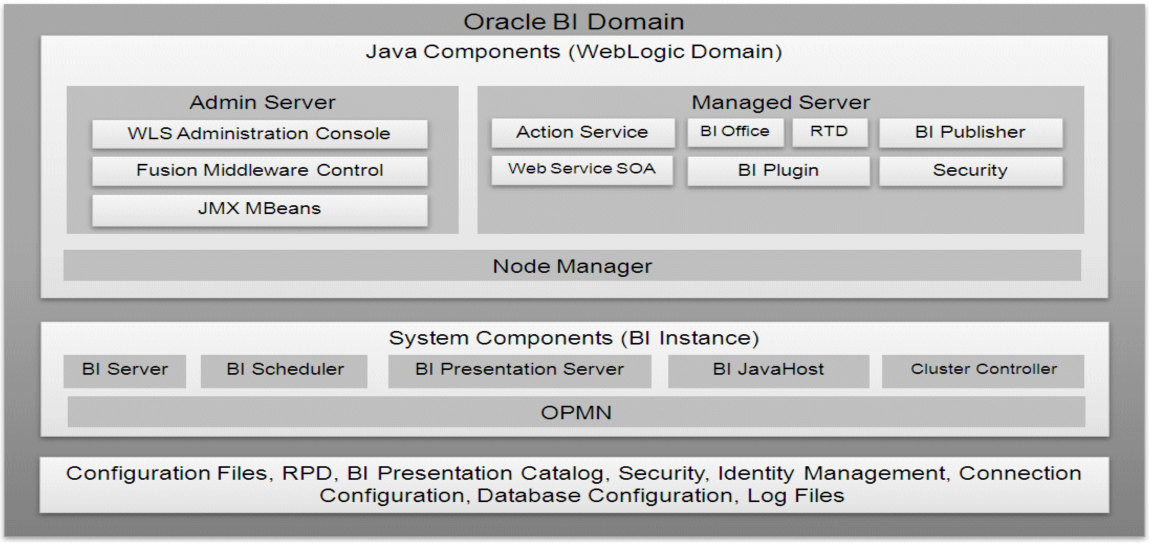
1.3 What is the Oracle Business Intelligence System Logical Architecture?

The Oracle Business Intelligence system logical architecture comprises a single integrated set of manageable components called the Oracle BI domain which can be installed and configured to work together on a single host or can be clustered across multiple hosts for performance and availability.

Oracle Business Intelligence on a single host

Figure 1–1 illustrates the Oracle Business Intelligence system architecture on a single host.

Figure 1–1 Oracle Business Intelligence System Logical Architecture (single host)


Oracle BI Domain
Java Components (WebLogic Domain)
Admin Server
WLS Administration Console
Fusion Middleware Control
JMX MBeans
Managed Server
Action Service
BI Office
RTD
BI Publisher
Web Service SOA
BI Plugin
Security
Node Manager
System Components (BI Instance)
BI Server
BI Scheduler
BI Presentation Server
BI JavaHost
Cluster Controller
OPMN
Configuration Files, RPD, BI Presentation Catalog, Security, Identity Management, Connection Configuration, Database Configuration, Log Files

BI Domain

A BI Domain consists of Java components that are deployed into one or more Java EE (JEE) containers within a single Weblogic domain; system (non-JEE) components and processes; and required configuration files, metadata repositories, and infrastructure.

Administration Server

A JEE container that runs in a dedicated Java virtual machine that contains Java components for administering the system.

For more information, see Section 1.5, "What is the Administration Server?".

Managed Server

A JEE container that runs in a dedicated Java virtual machine that provides the run-time environment for the Java-based services and applications within the system. An Oracle BI domain contains one or more Managed Servers that are distributed across one or more host computers.

Java Components

Java components are deployed as JEE applications to service SOAP, HTTP, and other forms of requests.

For more information about the Java components, see Chapter 1.4, "About Oracle BI Java Components and System Components".

Node Manager

Node Manager provides process management services for the Administration Server and Managed Server processes. For more information, see *Oracle Fusion Middleware Control Node Manager Administrator's Guide for Oracle WebLogic Server*.

System Components

System components are deployed as server processes and provide the core services that enable Oracle Business Intelligence.

For more information about the system components, see Section 1.4, "About Oracle BI Java Components and System Components.".

OPMN

OPMN (Oracle Process Manager and Notification server) maintains the Oracle Business Intelligence system component processes. For more information, see Section 1.6.3, "Oracle Process Manager and Notification Server (OPMN) Tool - Manage Oracle Business Intelligence System Components (for advanced users)"

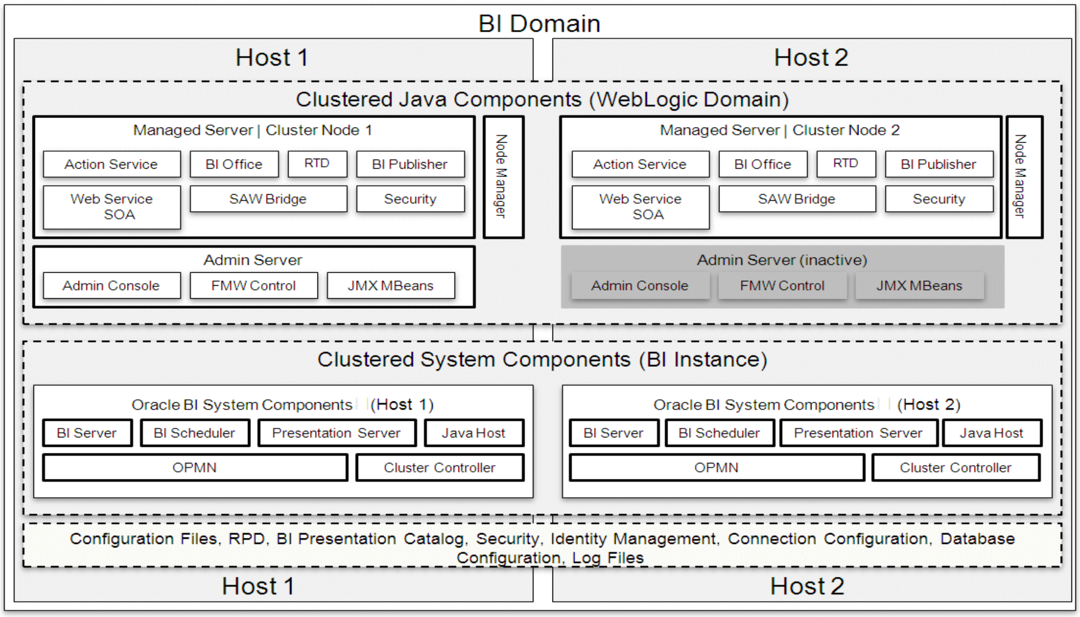
Other Oracle BI Domain Contents

The Oracle BI domain also contains all the necessary software, metadata, configuration files, RPD files, Oracle BI Presentation Catalog, connection and database configuration information that are required to run an Oracle Business Intelligence system.

Oracle Business Intelligence on multiple hosts

Oracle Business Intelligence can be installed and configured on multiple hosts. Figure 1–2 illustrates the system architecture on two hosts.

Figure 1–2 Oracle Business Intelligence System Logical Architecture (two hosts)



In Figure 1–2, the Java components (WebLogic domain) and the system components (BI instance) are clustered on each host as part of the single BI domain. The Administration Server exists on both hosts, but is only active on one host (in this example, Host 1).

1.4 About Oracle BI Java Components and System Components

The Oracle Business Intelligence Java and system components are listed as follows:

Java components — Deployed as one or more Java EE applications:

- **Administrative Components** — Enterprise Management applications and JMX MBeans for managing all configuration and run-time settings for Oracle Business Intelligence.
- **Oracle BI Publisher** — This component provides an enterprise reporting solution for authoring, managing, and delivering all types of highly formatted documents to employees, customers, and suppliers.

- **Oracle BI Office** — This component provides the integration between Oracle Business Intelligence and Microsoft Office products.
- **Oracle BI Action Services** — This component provides the dedicated Web services that are required by the Action Framework and that enable an administrator to manually configure which Web service directories can be browsed by users when they create actions.
- **Oracle Real-Time Decisions (Oracle RTD)** — This component provides enterprise analytics software solutions that enable companies to make better decisions in real time at key, high-value points in operational business processes.
- **Oracle BI Security Services** — This component provides dedicated Web services that enable the integration of the Oracle BI Server with the Oracle Fusion Middleware security platform.
- **Oracle BI SOA Services** — This component provides dedicated Web services for objects in the Oracle BI Presentation Catalog, to invoke analyses, agents, and conditions. They make it easy to invoke Oracle Business Intelligence functionality from Business Process Execution Language (BPEL) processes.
- **Oracle BI Plugin** — A JEE application that routes HTTP and SOAP requests to Oracle BI Presentation Services.

System components — Deployed as non-JEE components, such as processes and services written in C++ and J2SE:

- **Oracle BI Server** — This component provides the query and data access capabilities at the heart of Oracle Business Intelligence and provides services for accessing and managing the enterprise semantic model (stored in a file with a .RPD extension).
- **Oracle BI Presentation Services** — This component provides the framework and interface for the presentation of business intelligence data to Web clients. It maintains an Oracle BI Presentation Catalog service on the file system for the customization of this presentation framework.
- **Oracle BI Scheduler** — This component provides extensible scheduling for analyses to be delivered to users at specified times. (Oracle BI Publisher has its own scheduler.)
- **Oracle BI JavaHost** — This component provides component services that enable Oracle BI Presentation Services to support various components such as Java tasks for Oracle BI Scheduler, Oracle BI Publisher, and graph generation.
- **Oracle BI Cluster Controller** — This components distributes requests to the BI Server, ensuring requests are evenly load-balanced across all BI Server process instances in the BI domain.

For more information, see *Oracle Fusion Middleware Administrator's Guide*.

1.5 What is the Administration Server?

The Administration Server contains the administrative components that enable administration of a single or multi-node (that is, distributed) BI domain. Specifically the Administration Server contains the following

- **JMX MBeans** — Java components that provide programmatic access for managing a BI domain.
- **Fusion Middleware Control** — An administrative user interface that is used to manage the BI domain.

- **WebLogic Server Administration Console** — An adminstrative user interface that provides advanced management for Weblogic, JEE components, and security.

1.6 What System Administration Tools are Available for Managing Oracle Business Intelligence?

The following system administration tools are available to help you to manage Oracle Business Intelligence:

- Section 1.6.1, "Fusion Middleware Control - Monitor, Manage, and Configure Oracle Business Intelligence System Components"
- Section 1.6.2, "Oracle WebLogic Server Administration Console - Monitor and Manage Oracle Business Intelligence JEE Java Components"
- Section 1.6.3, "Oracle Process Manager and Notification Server (OPMN) Tool - Manage Oracle Business Intelligence System Components (for advanced users)"
- Section 1.6.4, "Oracle Weblogic Scripting Tool (WLST) - Programmatically Administer Oracle Business Intelligence"
- Section 1.6.5, "Oracle BI Administration Tool - Manage the Metadata Repository"

1.6.1 Fusion Middleware Control - Monitor, Manage, and Configure Oracle Business Intelligence System Components

Fusion Middleware Control is a browser-based tool and the recommended method for monitoring, managing, and configuring Oracle Business Intelligence components.

Fusion Middleware Control is used principally for managing the system components of a BI domain and provides support for the following:

- Starting, stopping, and restarting all system components and Managed Servers
- Configuring preferences and defaults
- Scaling out of system components
- Providing high availability configuration and status of system components
- Managing performance and monitoring system metrics
- Performing diagnostics and logging

Fusion Middleware Control also provides access to Oracle WebLogic Server Administration Console, where you monitor and manage Oracle Business Intelligence Java components.

Fusion Middleware Control is available only if the Administration Server is running, as described in Section 4.2, "Starting the Oracle Business Intelligence System."

For more information, see Chapter 2, "Managing Oracle Business Intelligence."

1.6.2 Oracle WebLogic Server Administration Console - Monitor and Manage Oracle Business Intelligence JEE Java Components

Oracle WebLogic Server is a Java EE application server that supports the deployment of Oracle Business Intelligence Java components in a robust, secure, highly available, and scalable environment.

For more information, see Chapter 2, "Managing Oracle Business Intelligence".

Oracle WebLogic Server Administration Console enables you to monitor and manage a WebLogic Server domain. Its capabilities include the following:

- Monitoring health and performance of JEE servers
- Configuring WebLogic domains
- Stopping and starting JEE servers
- Viewing JEE server logs

For more information, see Oracle Technology Network on `http://www.oracle.com/technology/index.html`.

1.6.3 Oracle Process Manager and Notification Server (OPMN) Tool - Manage Oracle Business Intelligence System Components (for advanced users)

OPMN is a process management tool that manages the Oracle Business Intelligence system components 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).

> **Note:** Fusion Middleware Control is the recommended approach for starting, stopping, and viewing the status of components. However, following a reboot on Linux you must use the opmnctl start, or startall command. OPMN is suitable only for advanced users.

OPMN provides the following functionality to manage the Oracle Business Intelligence system components:

- A command-line interface for advanced users to control Oracle Fusion Middleware components.

  For information, see Section 4.5, "Alternative Methods For Starting and Stopping Oracle Business Intelligence System Components"
- Automatic restart of processes when they become unresponsive or terminate unexpectedly.
- An integrated way to manage Oracle Fusion Middleware components.

1.6.4 Oracle Weblogic Scripting Tool (WLST) - Programmatically Administer Oracle Business Intelligence

The Oracle Weblogic Scripting Tool (WLST) is a command-line scripting environment (for advanced administrator use), which enables you to programmatically administer Oracle Business Intelligence. The WLST scripting environment is based on the Java scripting interpreter Jython. You can use this tool interactively on the command line, in batch scripts that are supplied in a file (Script Mode, where scripts invoke a sequence of WLST commands without requiring your input), or embedded in Java code. You can extend the WebLogic scripting language by following the Jython language syntax.

For more information, see:

- Section 30.2, "Accessing Oracle BI Systems Management API Methods"
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

1.6.5 Oracle BI Administration Tool - Manage the Metadata Repository

The Oracle BI Administration Tool enables you to manage the metadata repository. For information, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

1.7 Topics of Interest in Other Guides

Some topics that might be of interest to system administrators are covered in other guides. Table 1–2 lists these topics and indicates where to go for more information.

Table 1–2 Topics Covered in Other Guides

Topic	Where to Go for More Information
Third-party tools and relational data source adapters	Section 1.8, "System Requirements and Certification"
Configuring data sources	*Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*
Information about security, including configuring SSO and SSL	*Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
Installing and upgrading	*Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence* *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*

1.8 System Requirements and Certification

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

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

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

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

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

2 Managing Oracle Business Intelligence

This chapter introduces management and configuration of Oracle Business Intelligence using Oracle Enterprise Manager Fusion Middleware Control, Oracle WebLogic Server Administration Console, and the Fusion Middleware Control MBean Browser. The chapter includes the following topics:

- Section 2.1, "Why Use Fusion Middleware Control and WebLogic Server Administration Console?"
- Section 2.2, "Centrally Managing Oracle Business Intelligence Components Using Fusion Middleware Control"
- Section 2.3, "Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console"
- Section 2.4, "Managing Oracle Business Intelligence JMX MBeans Using the Fusion Middleware Control MBean Browser"

2.1 Why Use Fusion Middleware Control and WebLogic Server Administration Console?

You use Fusion Middleware Control and WebLogic Server Control Administration Console to centrally manage the Oracle Business Intelligence system. These Web-based tools support the most common system administration tasks for Oracle Business Intelligence. For more information, see Section 1.2, "Getting Started with Managing Oracle Business Intelligence."

Fusion Middleware Control enables you to manage system components by performing tasks such as monitoring status, starting and stopping processes, scaling out, resolving issues, and configuring components. You can also manage some aspects of Java components. For example, you can monitor their status and start and stop them.

WebLogic Server Administration Console enables you to monitor status and configure security for Java components. For information, see Chapter 1, "Introduction to Oracle Business Intelligence System Administration."

Locking Mechanism Enables Multiple Concurrent Administrators

With large deployments, you might have multiple administrators accessing the system concurrently to view the state of the system while other administrators might want to make configuration changes. Fusion Middleware Control and Oracle WebLogic Server prevent concurrent updates of the same configuration settings by multiple administrators by using a locking mechanism that allows only one administrator to make changes at any one time.

Note: Multiple administrators using the same administrator account could unknowingly make concurrent updates of the same configuration settings. It is therefore recommended that multiple administrator users do not share the same administrator account.

2.2 Centrally Managing Oracle Business Intelligence Components Using Fusion Middleware Control

You can use Fusion Middleware Control to centrally manage, monitor, and configure Oracle Business Intelligence system components (for example, the Oracle BI Server, Oracle BI Presentation Services, and Oracle BI Scheduler). You can also use Fusion Middleware Control to manage the Administration Server and Managed Servers.

This section contains the following topics:

- Section 2.2.1, "Logging into Fusion Middleware Control to Manage Oracle Business Intelligence"
- Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components"
- Section 2.2.3, "Using Fusion Middleware Control to Monitor Status and Start and Stop Oracle Business Intelligence Java Components"
- Section 2.2.4, "About the Navigation Tree in Fusion Middleware Control"

For more information about Fusion Middleware Control and how to use it, see *Oracle Fusion Middleware Administrator's Guide*.

2.2.1 Logging into Fusion Middleware Control to Manage Oracle Business Intelligence

To log in to Fusion Middleware Control, open a Web browser and enter the Fusion Middleware Control URL, in the following format:

```
http://hostname.domain:port/em
```

The port number is the number of the Administration Server, and the default port number is 7001.

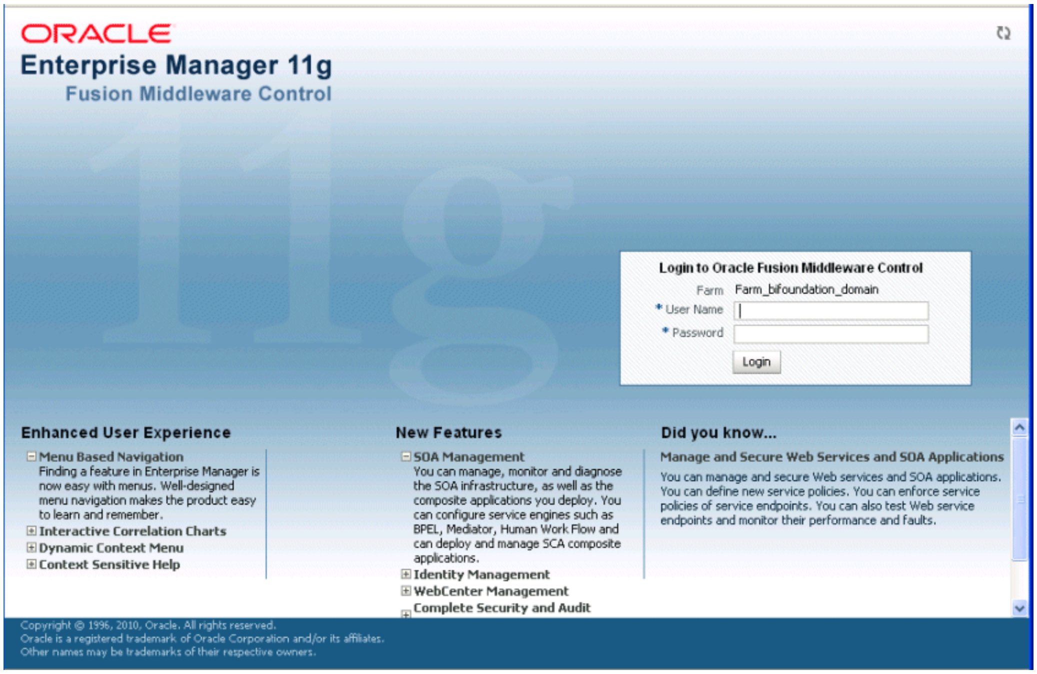
Fusion Middleware Control is available only if the Administration Server is running, as described in Section 4.2, "Starting the Oracle Business Intelligence System."

To log in to Fusion Middleware Control:

1. Enter the URL in a Web browser. For example:

   ```
   http://host1.example.com:7001/em
   ```

   The Fusion Middleware Control login page is displayed.



ORACLE Enterprise Manager 11g
Fusion Middleware Control

Login to Oracle Fusion Middleware Control

Farm Farm_bifoundation_domain
* User Name
* Password
Login

Enhanced User Experience

- **Menu Based Navigation**
  Finding a feature in Enterprise Manager is now easy with menus. Well-designed menu navigation makes the product easy to learn and remember.
- **Interactive Correlation Charts**
- **Dynamic Context Menu**
- **Context Sensitive Help**

New Features

- **SOA Management**
  You can manage, monitor and diagnose the SOA infrastructure, as well as the composite applications you deploy. You can configure service engines such as BPEL, Mediator, Human Work Flow and can deploy and manage SCA composite applications.
- **Identity Management**
- **WebCenter Management**
- **Complete Security and Audit**

Did you know...

Manage and Secure Web Services and SOA Applications

You can manage and secure Web services and SOA applications. You can define new service policies. You can enforce service policies of service endpoints. You can also test Web service endpoints and monitor their performance and faults.

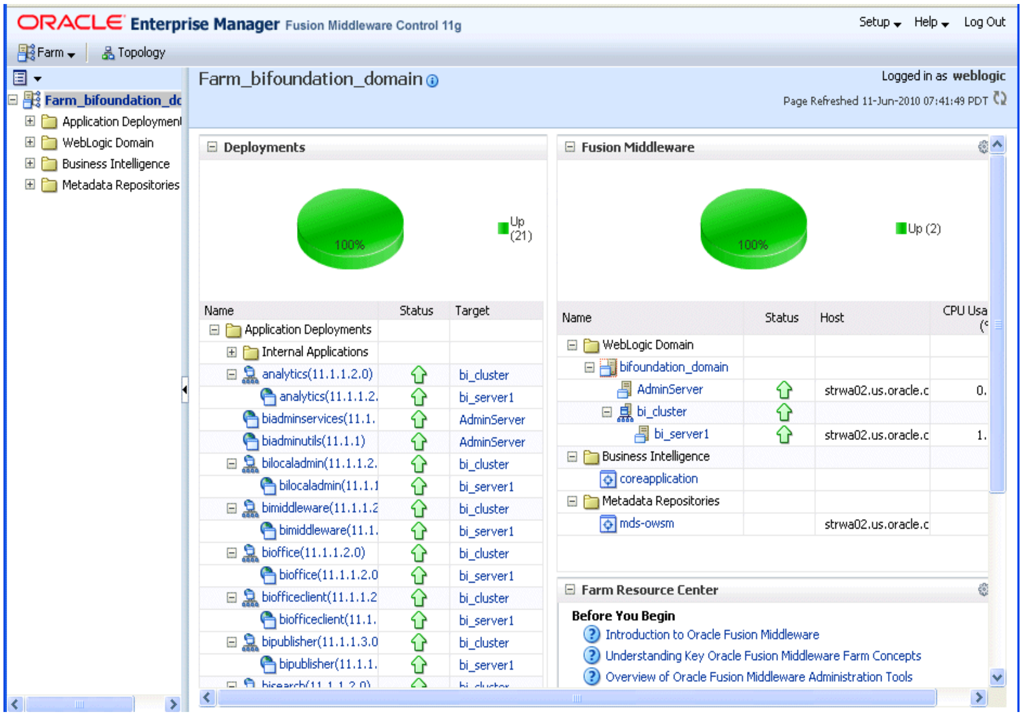
Copyright © 1996, 2010, Oracle. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

2. Enter the system administrator user name and password and click **Login**.

   This systemwide administration user name and password was specified during the installation process, and you can use it to log in to WebLogic Server Administration Console, Fusion Middleware Control, and Oracle Business Intelligence.

   Alternatively, enter any other user name and password that has been granted the Oracle BI Administrator application role.

   Fusion Middleware Control opens, as shown in the following figure.



ORACLE Enterprise Manager Fusion Middleware Control 11g

Farm_bifoundation_domain

Logged in as **weblogic**
Page Refreshed 11-Jun-2010 07:41:49 PDT

Deployments

Name	Status	Target
Application Deployments		
Internal Applications		
analytics(11.1.1.2.0)	⇧	bi_cluster
analytics(11.1.1.2.	⇧	bi_server1
biadminservices(11.1.	⇧	AdminServer
biadminutils(11.1.1)	⇧	AdminServer
bilocaladmin(11.1.1.2.	⇧	bi_cluster
bilocaladmin(11.1.1	⇧	bi_server1
bimiddleware(11.1.1.2	⇧	bi_cluster
bimiddleware(11.1.	⇧	bi_server1
bioffice(11.1.1.2.0)	⇧	bi_cluster
bioffice(11.1.1.2.0	⇧	bi_server1
biofficeclient(11.1.1.2	⇧	bi_cluster
biofficeclient(11.1.	⇧	bi_server1
bipublisher(11.1.1.3.0	⇧	bi_cluster
bipublisher(11.1.1.	⇧	bi_server1

Fusion Middleware

Name	Status	Host	CPU Usa ('
WebLogic Domain			
bifoundation_domain			
AdminServer	⇧	strwa02.us.oracle.c	0.
bi_cluster	⇧		
bi_server1	⇧	strwa02.us.oracle.c	1.
Business Intelligence			
coreapplication			
Metadata Repositories			
mds-owsm		strwa02.us.oracle.c	

Farm Resource Center

Before You Begin

- Introduction to Oracle Fusion Middleware
- Understanding Key Oracle Fusion Middleware Farm Concepts
- Overview of Oracle Fusion Middleware Administration Tools

> **Note:** If you have the browser configured to send HTTP requests to a proxy server, then you might have to configure the browser to not send Administration Server HTTP requests to the proxy server. If the Administration Server is on the same computer as the browser, then ensure that requests that are sent to localhost or 127.0.0.1 are not sent to the proxy server.

2.2.2 Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components

Use this topic to display Oracle Business Intelligence pages that enable you to manage the Oracle Business Intelligence system components:

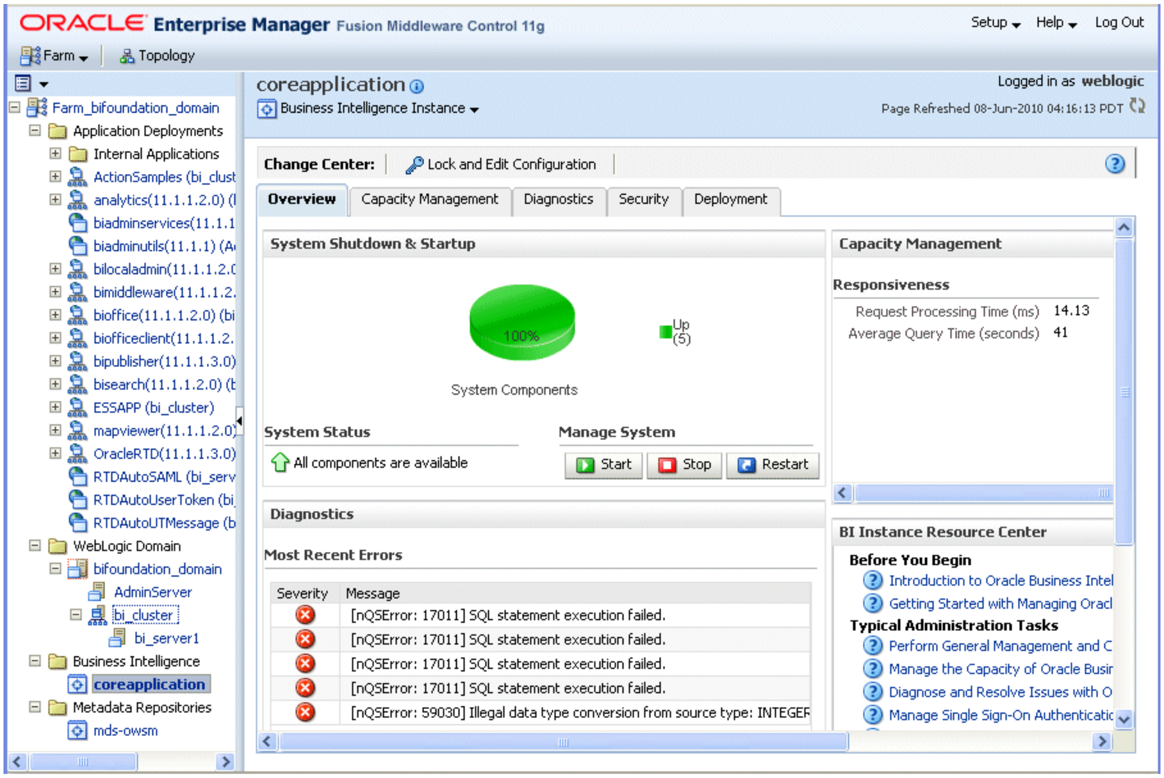
To manage Oracle Business Intelligence system components using Fusion Middleware Control:

1. Log in to Fusion Middleware Control.

   For more information, see Section 2.2.1, "Logging into Fusion Middleware Control to Manage Oracle Business Intelligence".

2. Expand the **Business Intelligence** folder and select the coreapplication node.

   Fusion Middleware Control displays the Overview page, as shown in the following figure.



coreapplication

Logged in as **weblogic**

Page Refreshed 08-Jun-2010 04:16:13 PDT

System Shutdown & Startup

System Status

All components are available

Manage System

Capacity Management

Responsiveness

Request Processing Time (ms)	14.13
Average Query Time (seconds)	41

Diagnostics

Most Recent Errors

Severity	Message
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 59030] Illegal data type conversion from source type: INTEGEF

BI Instance Resource Center

Before You Begin

- Introduction to Oracle Business Intel
- Getting Started with Managing Oracl

Typical Administration Tasks

- Perform General Management and C
- Manage the Capacity of Oracle Busir
- Diagnose and Resolve Issues with O
- Manage Single Sign-On Authenticatic

> **Note:** If the Business Intelligence folder is not visible or there is no coreapplication node under it, then Oracle Business Intelligence system components have not been installed. For information, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

The Overview page displays the current status of the system, by providing information about current availability, performance, and issues identified within the BI domain. (For more information, see Section 1.3, "What is the Oracle Business Intelligence System Logical Architecture?") The Overview page also enables you to start and stop Oracle Business Intelligence.

3. From the Overview page, select an appropriate tab to perform Oracle Business Intelligence management tasks.

   See Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings" for information about performing tasks on these tabs.

2.2.3 Using Fusion Middleware Control to Monitor Status and Start and Stop Oracle Business Intelligence Java Components

Use this topic to display the WebLogic Domain page, where you can monitor status and start and stop Oracle Business Intelligence Java components (Admin Server and Managed Servers).

You can also display the WebLogic Server Administration Console (using a link on the bifoundation_domain Summary page), where you can manage Oracle Business Intelligence Java components.

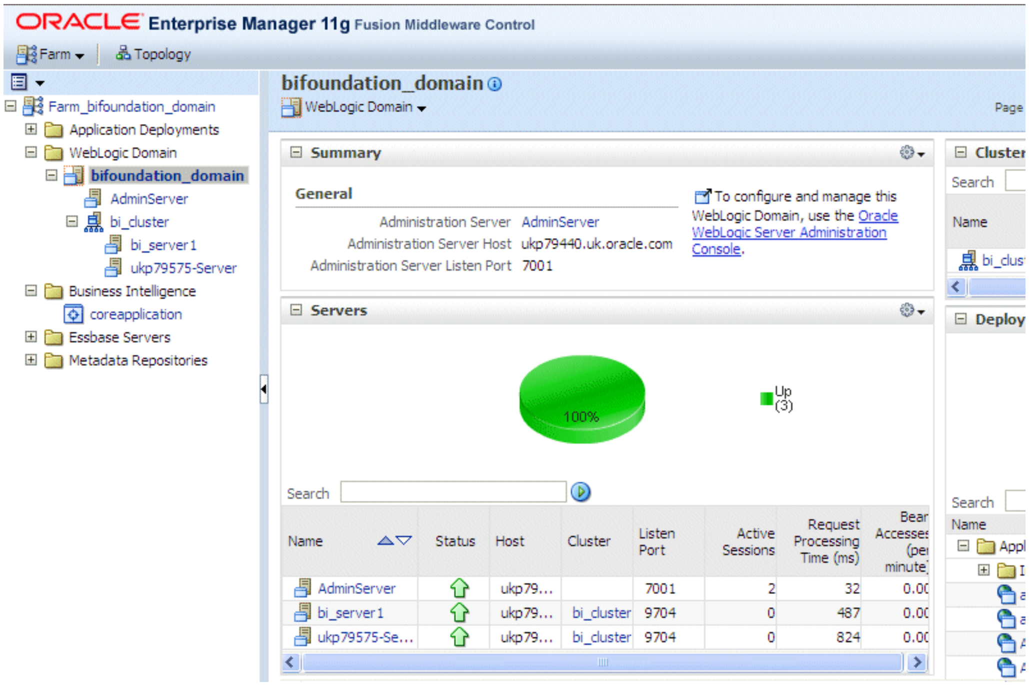
To manage Oracle Business Intelligence Java components using Fusion Middleware Control:

1. Log in to Fusion Middleware Control.

   For more information, see Section 2.2.1, "Logging into Fusion Middleware Control to Manage Oracle Business Intelligence."

2. Expand the **WebLogic Domain** folder and select the bifoundation_domain node.

   Fusion Middleware Control displays the bifoundation_domain page.



General

Administration Server AdminServer
Administration Server Host ukp79440.uk.oracle.com
Administration Server Listen Port 7001

To configure and manage this WebLogic Domain, use the Oracle WebLogic Server Administration Console.

Search

Name	Status	Host	Cluster	Listen Port	Active Sessions	Request Processing Time (ms)	Bear Accesses (per minute)
AdminServer	⇧	ukp79...		7001	2	32	0.00
bi_server1	⇧	ukp79...	bi_cluster	9704	0	487	0.00
ukp79575-Se...	⇧	ukp79...	bi_cluster	9704	0	824	0.00

The bifoundation_domain page is the starting point for monitoring status and for starting and stopping Oracle Business Intelligence Java components using Fusion Middleware Control. You can also click a link to display the WebLogic Server Administration Console, where you can manage and configure Oracle Business Intelligence Java components. For more information, see Section 2.3, "Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console."

3. Using the bifoundation_domain page, you can perform the following Oracle Business Intelligence management tasks:

   - View the status of Admin Server and Managed Servers (bi_server<*n*>).
   - Start and stop selected Java components (AdminServer or bi_server<*n*>) using the WebLogic Domain menu Control option.

     For information, see, Section 4.8, "Confirming Whether the Managed Server is Running, and Starting it if Necessary."
   - Manage or configure the WebLogic domain in WebLogic Server Administration Console by clicking a link on the WebLogic Domain page.

     For information about using WebLogic Server Administration Console, see Section 2.3, "Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console."

2.2.4 About the Navigation Tree in Fusion Middleware Control

The navigation tree enables you to navigate within the BI domain that can be managed by Fusion Middleware Control.

Depending on the choices made during installation, the following domain components can be displayed as nodes in the navigation tree:

- **Application Deployments**

The Application Deployments node shows all the applications that are deployed into the BI domain (for example, analytics, Oracle Business Intelligence Add-in for Microsoft Office, Oracle BI Publisher, and Oracle Real-Time Decisions).

- **WebLogic Domain**

  These nodes display summary information for the WebLogic server. Select a node and click the **Oracle WebLogic Server Administration Console** link to display the WebLogic Server Administration Console, where you can administer Oracle WebLogic Server.

  – **bifoundation_domain**

    This node represents the WebLogic domain for Oracle Business Intelligence with an AdminServer node that contains the Administration Server and a bi_cluster node that contains Managed Servers (a single node cluster by default). For information, see Section 1.5, "What is the Administration Server?"

    * **AdminServer**
    * **bi_cluster**

- **Business Intelligence**

  – **coreapplication**

    This node represents the Oracle Business Intelligence system components that can be managed using Fusion Middleware Control.

    Select this node to display the Overview page and manage the system components.

- **Metadata Repositories**

  This node represents the Metadata Services (MDS) schema repositories that can be managed using Fusion Middleware Control.

2.3 Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console

You use the Oracle WebLogic Server Administration Console to centrally manage Oracle Business Intelligence Java components.

You display Oracle WebLogic Server Administration Console, using one of the following methods:

- Clicking a link on the Overview page in Fusion Middleware Control
- Entering a URL into a Web browser window

The Oracle WebLogic Server Administration Console is available only if the Administration Server for WebLogic Server is running. For information, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

To display Oracle WebLogic Server Administration Console:

1. If the Administration Server for WebLogic Server is not running, start it.

   For information, see Section 4.7, "Starting and Stopping the Administration Server and all Oracle Business Intelligence Java Components on Windows and UNIX."

2. Display the Oracle WebLogic Server Administration Console using one of the following methods:

   Clicking a link on the Overview page in Fusion Middleware Control:

a. Display Fusion Middleware Control. For information, see Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."

b. Expand the WebLogic Domain node and select the bifoundation_domain.

c. Click the **Oracle WebLogic Server Administration Console** link in the Summary region.

The Oracle WebLogic Server Administration Console login page is displayed.

Using a URL in a Web browser window:

a. Start a Web browser.

b. Enter the following URL into the browser:

http://<*host*>:<*port*>/console/

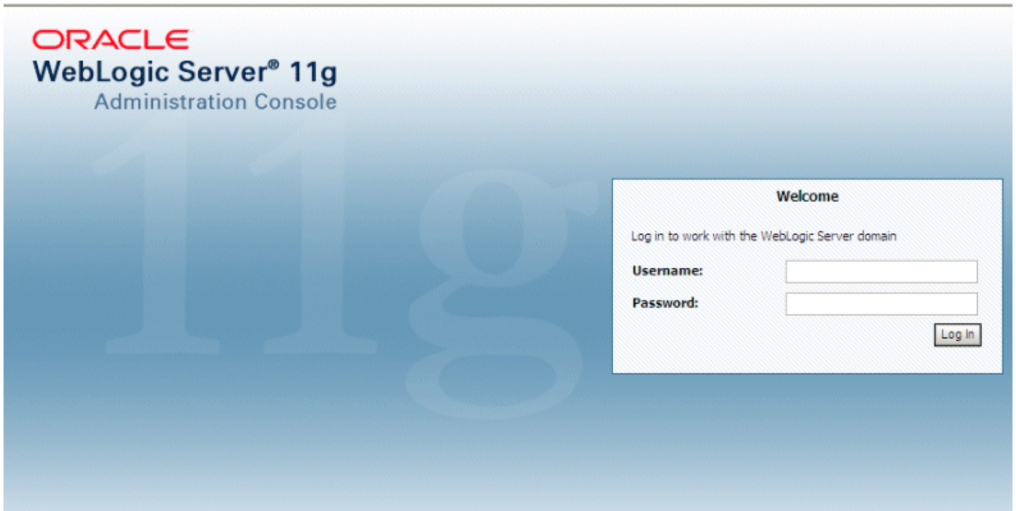
For example, http://mycomputer:7001/console/

where `host` is the DNS name or IP address of the Administration Server and `port` is the listen port on which the Administration Server is listening for requests (port 7001 by default). If you have configured a domain-wide Administration port, then use that port number. If you configured the Administration Server to use Secure Socket Layer (SSL), then you must add the letter 's' after http as follows:

https://<*host*>:7001/console/

The above URL example uses SSL.

The Oracle WebLogic Server Administration Console login page is displayed, as shown in the following figure.



3. Enter the system administrator user name and password and click **Login**.

This systemwide administration user name and password was specified during the installation process, and you can use it to log in to WebLogic Server Administration Console, Fusion Middleware Control, and Oracle Business Intelligence. Alternatively, enter a user name that belongs to one of the following security groups:

- Administrators
- Operators

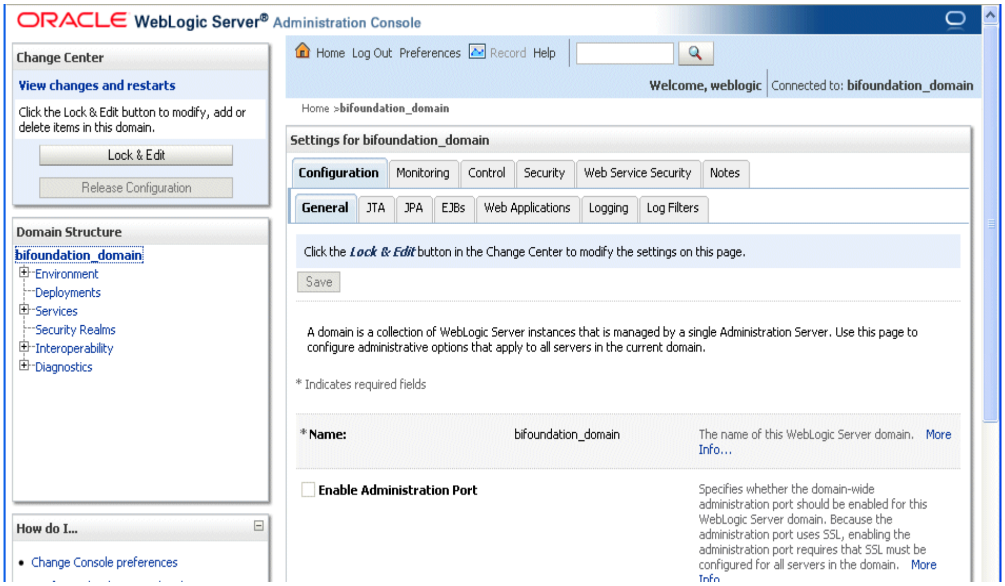
- Deployers
- Monitors

These groups provide various levels of access to system administration functions in the Oracle WebLogic Server Administration Console.

Using the security system, you can add or delete users to one of these groups to provide controlled access to the Console.

If you have the browser configured to send HTTP requests to a proxy server, then you might have to configure the browser to not send Administration Server HTTP requests to the proxy. If the Administration Server is on the same computer as the browser, then ensure that requests sent to localhost or 127.0.0.1 are not sent to the proxy.

In Oracle WebLogic Server Administration Console you select the bifoundation_ domain page, as shown in the following figure.



Click the Lock & Edit button to modify, add or delete items in this domain.

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

A domain is a collection of WebLogic Server instances that is managed by a single Administration Server. Use this page to configure administrative options that apply to all servers in the current domain.

* Indicates required fields

* Name:	bifoundation_domain	The name of this WebLogic Server domain. More Info...
□ Enable Administration Port		Specifies whether the domain-wide administration port should be enabled for this WebLogic Server domain. Because the administration port uses SSL, enabling the administration port requires that SSL must be configured for all servers in the domain. More Info...

You can monitor and manage Oracle Business Intelligence Java components from this page.

For more information, see the *Oracle WebLogic Server Administration Console Help* system.

2.4 Managing Oracle Business Intelligence JMX MBeans Using the Fusion Middleware Control MBean Browser

This section introduces the Fusion Middleware Control MBean Browser and explains how to display it.

- Section 2.4.1, "What is the Fusion Middleware Control MBean Browser?"
- Section 2.4.2, "Displaying the Fusion Middleware Control MBean Browser"

2.4.1 What is the Fusion Middleware Control MBean Browser?

The Fusion Middleware Control MBean Browser is an Oracle Web application (based on JMX MBean containers), that can be used instead of Fusion Middleware Control to perform centralized and local management of Oracle Business Intelligence system components.

2.4.2 Displaying the Fusion Middleware Control MBean Browser

This section describes how to display the Fusion Middleware Control MBean Browser.

Although you can use the Fusion Middleware Control MBean Browser to update Oracle Business Intelligence configuration settings, the same settings are more conveniently accessible using the Oracle Business Intelligence pages in Fusion Middleware Control. You display these pages by selecting the coreapplication node in the Business Intelligence folder as described in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To display the Fusion Middleware Control MBean Browser:

1. Display Fusion Middleware Control.

   For information, see Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."

2. In the Navigator window, expand the WebLogic Domain folder and the bifoundation_domain node, and select the AdminServer node.

   For information, see Section 1.5, "What is the Administration Server?"

3. Display the WebLogic Server menu and select **System MBean Browser** from the menu to display the Fusion Middleware Control System MBean Browser.

   For information about how to access Oracle Business Intelligence MBeans, see Section 3.5, "Using the Fusion Middleware Control MBean Browser to Update Configuration Settings.".

4. Make and save your changes.

3 Configuring the Oracle Business Intelligence System

This chapter explains how you configure the Oracle Business Intelligence system and provides information about the location of configuration files.

This chapter contains the following topics:

3.1 Configuring Oracle Business Intelligence

Oracle recommends using Fusion Middleware Control to configure Oracle Business Intelligence. However, if configuration settings cannot be changed using this method, then you can use the Oracle BI Systems Management API or the Fusion Middleware Control MBean Browser. You can also manually edit configuration files if you take the appropriate precautions.

Table 3–1 describes the methods that you can use for configuring Oracle Business Intelligence. All the methods result in changes being made in one or more configuration files.

Table 3–1 Methods for Configuring Oracle Business Intelligence

What to you want to do?	What tools can you use?	How are updates made?
Centrally manage configuration settings	- Fusion Middleware Control For information, see Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings." - Fusion Middleware Control MBean Browser For information, see Section 3.5, "Using the Fusion Middleware Control MBean Browser to Update Configuration Settings."	Automatically, through Oracle BI Systems Management API methods. For information, see Chapter 30, "Introducing the Oracle BI Systems Management API."
Programmatically change configuration settings	- WLST (WebLogic Scripting Tool) - Java Program For information, see Section 3.3, "Using the Oracle BI Systems Management API to Update Oracle Business Intelligence Configuration Settings."	Using scripts or programs to programmatically access Oracle BI Systems Management API methods.
Manually change configuration settings	- Text editor For information, see Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings." **Caution:** Settings managed by Fusion Middleware Control must not be manually updated, and are commented as such in configuration files. **Caution:** Changes must be rolled out correctly in configuration files in all nodes in a cluster to avoid system damage.	Manually, using a text editor to enter changes in configuration files.

3.2 Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings

Using Fusion Middleware Control to update Oracle Business Intelligence configuration settings is the recommended method, because it manages the distribution of changes across the system, and across all computers if the system is clustered.

To update Oracle Business Intelligence configuration settings using Fusion Middleware Control:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Select the appropriate page and tabs to display the settings to change.
3. Click **Lock and Edit Configuration** to enable changes to be made.

> **Caution:** Multiple administrators should not share the same administrator account, as they could unknowingly make concurrent updates to the same configuration settings.

4. Make the appropriate changes on each page.
5. Click **Apply** on each page after you have made your changes.
6. When you have finished making your changes, do one of the following:

- Click **Activate Changes** to execute your changes and release the lock to enable another system administrator to make changes.
- Click **Release Configuration** to undo all changes you made since clicking **Lock and Edit Configuration** and release the lock to enable another system administrator to make changes.

7. After you have activated your changes, go to the Overview page and click **Restart**.

3.3 Using the Oracle BI Systems Management API to Update Oracle Business Intelligence Configuration Settings

You can use the Oracle BI Systems Management API to update Oracle Business Intelligence configuration settings that are normally managed by Fusion Middleware Control. For information, see Section 30.3.3, "Updating Configuration Settings Using the Oracle BI Systems Management API."

3.4 Using a Text Editor to Update Oracle Business Intelligence Configuration Settings

You can manually update configuration settings that are not normally managed by Fusion Middleware Control using a text editor.

> **Caution:** If you manually update configuration settings that are not normally managed by Fusion Middleware Control, then you must likely make identical changes in multiple copies of configuration files on multiple computers. If you fail to make changes in all of the necessary configuration files, then the system might be damaged.

You can use a text editor to manually update Oracle Business Intelligence configuration settings that are not normally managed by Fusion Middleware Control, as described in the following procedure.

To manually update Oracle Business Intelligence configuration settings that are not normally managed by Fusion Middleware Control:

1. Make a backup copy of the files that you plan to edit.
2. Open the configuration file in a text editor.

   > **Note:** If Oracle Business Intelligence is installed on multiple computers or in multiple locations, then a copy of the configuration file exists for each installation, and each file contains duplicate settings. Therefore, **you must locate and make the same changes in each copy of a configuration file**.

   For information, see Section 3.6, "Where are Configuration Files Located?"
3. In each configuration file to change, locate the element or create the new element if a setting must be added to the file.
4. Enter the appropriate changes.
5. Save your changes and close the configuration file.
6. Restart Oracle Business Intelligence.

For information, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

It is possible to manually update configuration settings that are normally managed by Fusion Middleware Control. However, you should never attempt such updates unless you are specifically instructed to do so by Oracle Support Services. For information, see Appendix H, "Manually Updating Fusion Middleware Control Settings."

3.5 Using the Fusion Middleware Control MBean Browser to Update Configuration Settings

Use Fusion Middleware Control to update configuration settings, unless specifically directed to use the MBean Browser. For information, see Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To update configuration settings using the Fusion Middleware Control MBean Browser:

1. Display the Fusion Middleware Control MBean Browser.

   For information, see Section 2.4.2, "Displaying the Fusion Middleware Control MBean Browser."

2. Lock the domain.

   Before you change any Oracle Business Intelligence-related configuration attribute settings using the Fusion Middleware Control MBean Browser, you must lock the domain as follows:

   a. Navigate to the following MBean (the first BIDomain MBean): oracle.biee.admin, Domain:bifoundation_domain, BIDomain, BIDomain.

   b. Display the Operations tab.

   c. Click the **Lock operation** link.

      When you have locked the domain, you can start to update settings.

3. Expand a folder to display the details for a group of configuration file attributes.

   For example, to configure Oracle Business Intelligence e-mail settings, expand the BIDomain.BIInstance.EmailConfiguration folder, and select the underlying node of the same name.

   To configure a different group of Oracle Business Intelligence configuration settings, expand the required folder, and select it's underlying node.

4. Create or update a value in one of the following ways:

   – Entering directly into a field in the Attributes tab:

     * Enter a new value into the **Value** field for the appropriate row.

     * Click **Apply** to save the changes, or **Revert** to undo the changes.

   – Entering into a new page that you display from the Attributes tab:

     * Click the link in the **Name** column to display a new page.

     * Enter a new value in the **Value** field.

     * Click **Apply** to save the changes, or **Revert** to undo the changes.

     * Click **Return** to close the page.

5. Click **Revert** to undo your changes.

6. Repeat the previous steps to create, or change other settings.
7. Release the lock on the domain.

   After you have made changes you must release the lock on the domain by committing, or rolling back your changes as follows:

   a. Navigate to the following MBean: oracle.biee.admin, Domain:bifoundation_domain, BIDomain, BIDomain.
   b. Display the Operations tab.
   c. (Optional) Click the commit operation link to save your changes and release the lock on the domain.

      Use this option when you are satisfied with the changes and you want to release the lock on the domain.
   d. (Optional) Click the rollback operation link to undo your changes and release the lock on the domain.

      Use this option when you do not want to commit your changes and want to release the lock on the domain.

3.6 Where are Configuration Files Located?

Oracle Business Intelligence configuration files, used to configure the behavior of the system, are found in various locations within the Oracle Fusion Middleware Home.

Each host has its own set of configuration files for the components that are deployed on that host.

Oracle Business Intelligence system component configuration files can be found under:

MW_HOME\ORACLE_INSTANCE\\config\COMPONENT\bi_component_name\\

For example, the configuration file for the Oracle BI Server might be located in:

D:\mw_home\instances\instance1\config\OracleBIServerComponent\
coreapplication_obis1

Table 3–2 lists the names and locations of configuration files for each Oracle Business Intelligence system component. You are advised to use Fusion Middleware Control to update Oracle Business Intelligence configuration settings and ensure that changes are correctly propagated. For information, see Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

For information about diagnostic log configuration files, see Section 8.2.2, "What Are Diagnostic Log Configuration Files and Where are They Located?"

Table 3–2 Oracle Business Intelligence System Component Configuration File Locations

BI Component	Configuration File	File Location
Oracle BI Server	NQSConfig.INI logconfig.xml	*ORACLE_INSTANCE*\config\OracleBIServerComponent\coreapplication_obis*n* For example: \instances\instance1\config\OracleBIServerComponent\coreapplication_obis1 **Note:** Although DBFeatures.ini is also located in this directory, do not edit this file directly. See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about how to edit features for a database.
Oracle BI Presentation Services	instanceconfig.xml	*ORACLE_INSTANCE*\config\OracleBIPresentationServicesComponent\coreapplication_obips*n* For example: \instances\instance1\config\OracleBIPresentationServicesComponent\coreapplication_obips1 **Note:** Do not add elements to the instanceconfig.xml file unless you are overriding the stated default values. Override only those settings that are necessary for configuring the system to meet the needs of your organization.
Cluster Controller	NQClusterConfig.INI ccslogging.xml	*ORACLE_INSTANCE*\config\OracleBIApplication\coreapplication For example: \instances\instance1\config\OracleBIApplication\coreapplication
Oracle BI Scheduler	instanceconfig.xml ccslogging.xml (for Cluster Controller)	*ORACLE_INSTANCE*\config\OracleBISchedulerComponent\coreapplication_obisch*n* For example: \instances\instance1\config\OracleBISchedulerComponent\coreapplication_obisch1
JavaHost	config.xml logging_config.xml	*ORACLE_INSTANCE*\config\OracleBIJavaHostComponent\coreapplication_obijh*n* For example: \instances\instance1\config\OracleBIJavaHostComponent\coreapplication_obijh1
Oracle BI Presentation Services Plug-in	bridgeconfig.properties	*MW_HOME*\user_projects\domains\domain_name\config\fmwconfig\biinstances\coreapplication For example: mw_home\user_projects\domains\bifoundation_domain\config\fmwconfig\biinstances\coreapplication

For more information about Oracle Business Intelligence installations, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*

Part II

Managing Processes and Components

This part explains how to manage processes and components in Oracle BI Enterprise Edition. It contains the following chapter:

4 Starting and Stopping Oracle Business Intelligence

This chapter explains how to start and stop Oracle Business Intelligence. It contains the following topics:

4.1 About Starting and Stopping Oracle Business Intelligence

System administrators start and stop the Oracle Business Intelligence system and its components to perform a range of maintenance operations that require process downtime. Understanding the state (that is, up, down, starting, and stopping) of each component in the Oracle Business Intelligence system is an essential activity when diagnosing and resolving availability and performance issues, and when performing life-cycle and management operations.

Oracle Business Intelligence runs within Oracle WebLogic Server, and therefore Oracle WebLogic Server must be started before Oracle Business Intelligence components can be started and maintained.

To make changes to server configuration settings, the Oracle BI Presentation Catalog, the Repository (.rpd file offline), and other settings, you must restart the appropriate Oracle Business Intelligence components before those changes can take effect.

When you stop Oracle Business Intelligence, end users are logged out, and when ready, the system prompts you to log in again, ensuring session state consistency.

4.2 Starting the Oracle Business Intelligence System

To start the Oracle Business Intelligence system, you first start the Administration Server and Managed Servers, and then you start the system components.

If the computer that hosts the Administration Server is not running or has been rebooted, then you must start the Oracle Business Intelligence system.

The following conditions must be met to start the Oracle Business Intelligence system:

- The RPD file must be closed (if using the Administration Tool) so that it does not hold the read/write lock.

  For information, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

- The repository database (which contains Scheduler schemas) that was specified during installation must be running, and a network connection to it must be available. Otherwise, error messages are displayed.

To start the Oracle Business Intelligence system:

1. Start the Administration Server (which includes Oracle Business Intelligence Java components).

   For information, see Section 4.7, "Starting and Stopping the Administration Server and all Oracle Business Intelligence Java Components on Windows and UNIX."

2. Start Oracle WebLogic Node Manager manually as follows:

   On Windows use the **Programs** > **Oracle Business Intelligence** > **Start BI Services** on the Start menu. This also starts the OPMN Service. (Node Manager normally starts automatically on Windows.)

   On UNIX, navigate to:

   /*<MW_HOME>*/wlserver_10.3/server/bin

   In a software-only installation, a directory other than /wls_server_10.3 might be used.

   Enter the following command (which requires the administrator user name and password):

   ```
   ./startNodeManager.sh
   ```

   You can also start Node Manager using a script. For information, see *Oracle Fusion Middleware Control Node Manager Administrator's Guide for Oracle WebLogic Server*.

3. Use the Oracle WebLogic Server Administration Console to check whether the Managed Server is running, and start it if necessary.

   Note: In order to start the Managed Server using the WebLogic Server Administration Console, the Node Manager must be running.

   For information, see Section 4.8, "Confirming Whether the Managed Server is Running, and Starting it if Necessary."

4. Start system components.

   - Using Fusion Middleware Control

For information, see Section 4.3, "Using Fusion Middleware Control to Start and Stop Oracle Business Intelligence System Components and Java Components."

- Using the Windows Services

  From the Windows Start menu, select **Programs**, **Oracle Business Intelligence**, and **Start BI Services** (starts OPMN and Node Manager services)

  For information, see Section 4.5.2, "Using a Windows Service to Start and Stop Oracle Business Intelligence System Components."

- Using the opmnctl command line

  For information, see Section 4.5.1, "Using the OPMN Command Line to Start, Stop, Restart, and View the Status of Oracle Business Intelligence System Components."

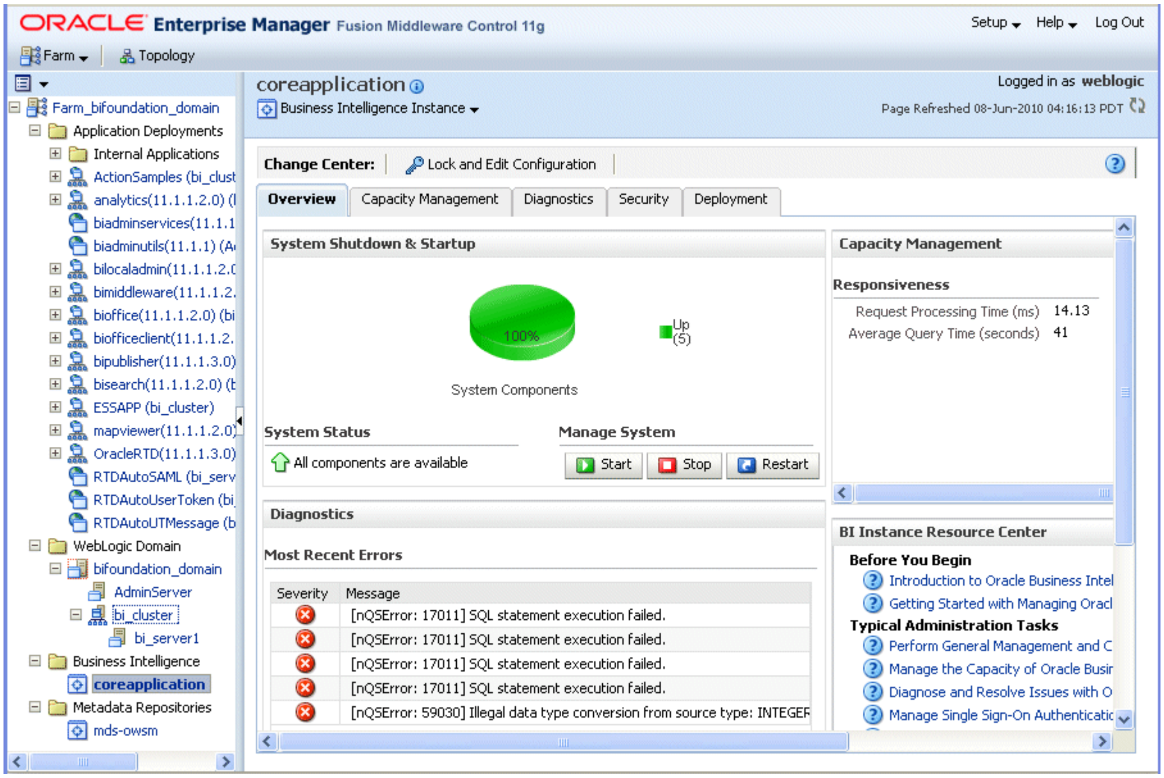
4.3 Using Fusion Middleware Control to Start and Stop Oracle Business Intelligence System Components and Java Components

If the Oracle Business Intelligence system has been started, then you can start, stop, and restart the Oracle Business Intelligence system, including selected system components and Java components using Fusion Middleware Control.

If Fusion Middleware Control is not available, then see Section 4.6, "Confirming Whether the Administration Server is Running."

To start, stop, and restart Oracle Business Intelligence system components and Java components using Fusion Middleware Control:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."



- Farm_bifoundation_domain
  - Application Deployments
    - Internal Applications
    - ActionSamples (bi_clust
    - analytics(11.1.1.2.0) (I
    - biadminservices(11.1.1
    - biadminutils(11.1.1) (A
    - bilocaladmin(11.1.1.2.(
    - bimiddleware(11.1.1.2.
    - bioffice(11.1.1.2.0) (bi
    - biofficeclient(11.1.1.2.
    - bipublisher(11.1.1.3.0)
    - bisearch(11.1.1.2.0) (t
    - ESSAPP (bi_cluster)
    - mapviewer(11.1.1.2.0)
    - OracleRTD(11.1.1.3.0)
    - RTDAutoSAML (bi_serv
    - RTDAutoUserToken (bi
    - RTDAutoUTMessage (b
  - WebLogic Domain
    - bifoundation_domain
      - AdminServer
      - bi_cluster
        - bi_server1
  - Business Intelligence
    - coreapplication
  - Metadata Repositories
    - mds-owsm

coreapplication

Business Intelligence Instance

Logged in as weblogic

Page Refreshed 08-Jun-2010 04:16:13 PDT

Change Center: Lock and Edit Configuration

System Shutdown & Startup

System Status: All components are available

Manage System: Start Stop Restart

Capacity Management

Responsiveness

Request Processing Time (ms)	14.13
Average Query Time (seconds)	41

Diagnostics

Most Recent Errors

Severity	Message
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 17011] SQL statement execution failed.
☒	[nQSError: 59030] Illegal data type conversion from source type: INTEGER

BI Instance Resource Center

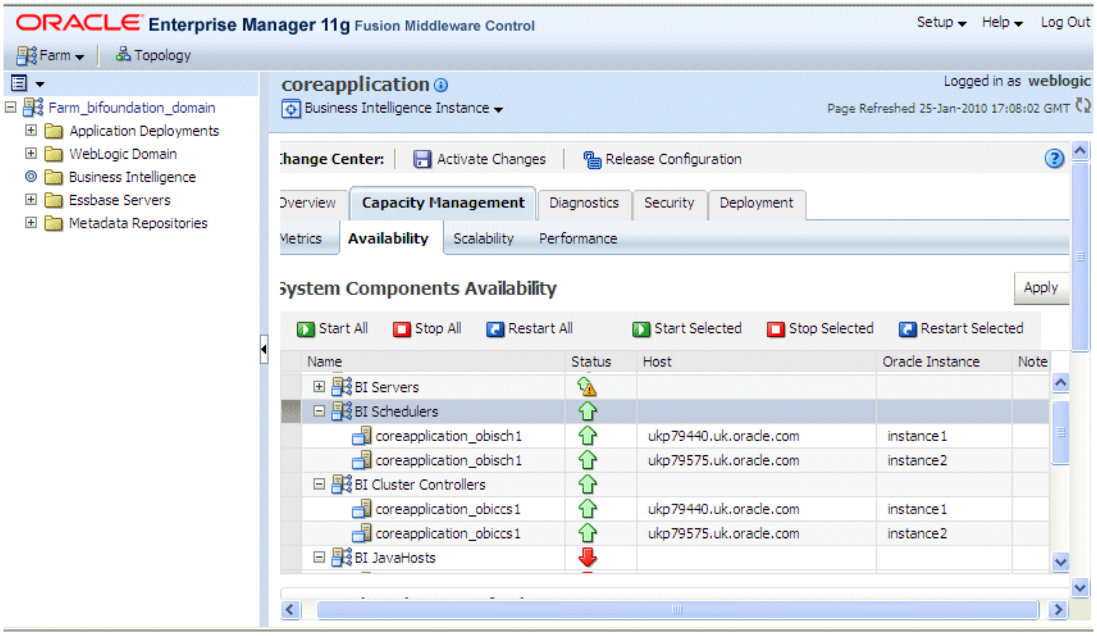
Before You Begin
- Introduction to Oracle Business Intel
- Getting Started with Managing Oracl

Typical Administration Tasks
- Perform General Management and C
- Manage the Capacity of Oracle Busir
- Diagnose and Resolve Issues with O
- Manage Single Sign-On Authenticatic

2. Use the buttons in the System Shutdown & Startup area to start, stop, or restart the entire Oracle Business Intelligence system.

   Restart stops the system before starting it again.

3. To start, stop, or restart individual Oracle Business Intelligence system components, display the Availability tab of the Capacity Management page, then select a process for a selected server and use the appropriate button to start, stop, or restart individual system components as appropriate.



- Farm_bifoundation_domain
  - Application Deployments
  - WebLogic Domain
  - Business Intelligence
  - Essbase Servers
  - Metadata Repositories

coreapplication

Business Intelligence Instance

Logged in as weblogic

Page Refreshed 25-Jan-2010 17:08:02 GMT

hange Center: Activate Changes Release Configuration

System Components Availability

Name	Status	Host	Oracle Instance	Note
BI Servers	⚠			
BI Schedulers	↑			
coreapplication_obisch1	↑	ukp79440.uk.oracle.com	instance1	
coreapplication_obisch1	↑	ukp79575.uk.oracle.com	instance2	
BI Cluster Controllers	↑			
coreapplication_obiccs1	↑	ukp79440.uk.oracle.com	instance1	
coreapplication_obiccs1	↑	ukp79575.uk.oracle.com	instance2	
BI JavaHosts	↓			

You can use other methods to start and stop Oracle Business Intelligence system and Java components. For more information, see:

- Section 4.4, "Using Oracle WebLogic Server Administration Console to Start and Stop Java Components"
- Section 4.5, "Alternative Methods For Starting and Stopping Oracle Business Intelligence System Components"

4.4 Using Oracle WebLogic Server Administration Console to Start and Stop Java Components

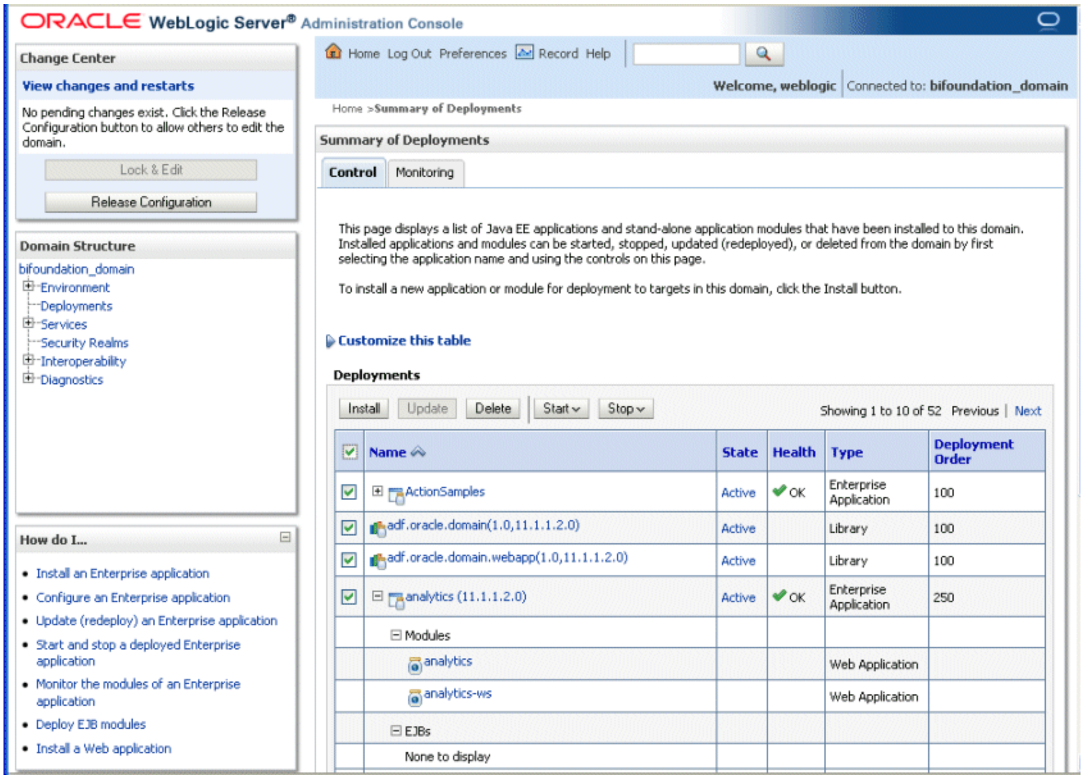
You can use Oracle WebLogic Server Administration Console to start and stop Java components. You can also use Fusion Middleware Control to start and stop the Java components, as described in Section 4.3, "Using Fusion Middleware Control to Start and Stop Oracle Business Intelligence System Components and Java Components."

To use the Oracle WebLogic Server Administration Console to start and stop Java components:

1. Start the Oracle WebLogic Server Administration Console.

   For more information, see Section 2.3, "Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console."

2. In the Domain Structure region, click **Deployments**.
3. The Oracle WebLogic Server Administration Console displays the Summary of Deployments page.
4. Display the Control tab.
5. Select a check box for each component to start or stop.
6. Click **Start** or **Stop** to start or stop the selected components as required.


ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic
Connected to: bifoundation_domain
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
bifoundation_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Install an Enterprise application
Configure an Enterprise application
Update (redeploy) an Enterprise application
Start and stop a deployed Enterprise application
Monitor the modules of an Enterprise application
Deploy EJB modules
Install a Web application
Home >Summary of Deployments
Summary of Deployments
Control
Monitoring
Customize this table
Install
Update
Delete
Start
Stop
Previous
Next

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Deployments

Showing 1 to 10 of 52

☑	Name	State	Health	Type	Deployment Order
☑	⊞ ActionSamples	Active	✓ OK	Enterprise Application	100
☑	adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
☑	adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
☑	⊟ analytics (11.1.1.2.0)	Active	✓ OK	Enterprise Application	250
	⊟ Modules				
	analytics			Web Application	
	analytics-ws			Web Application	
	⊟ EJBs				
	None to display				

4.5 Alternative Methods For Starting and Stopping Oracle Business Intelligence System Components

You can start and stop Oracle Business Intelligence system components using the methods that are described in the following sections:

- Section 4.5.1, "Using the OPMN Command Line to Start, Stop, Restart, and View the Status of Oracle Business Intelligence System Components"
- Section 4.5.2, "Using a Windows Service to Start and Stop Oracle Business Intelligence System Components"
- Section 4.5.3, "Configuring OPMN to Start and Stop Oracle Business Intelligence System Components Automatically on Linux"
- Section 4.5.4, "Using the Oracle BI Systems Management API to Programmatically Start and Stop Oracle Business Intelligence"

4.5.1 Using the OPMN Command Line to Start, Stop, Restart, and View the Status of Oracle Business Intelligence System Components

You can control Oracle Business Intelligence system components across a network using opmnctl commands. For information about OPMN, see Section 1.6.3, "Oracle Process Manager and Notification Server (OPMN) Tool - Manage Oracle Business Intelligence System Components (for advanced users)."

To start, stop, restart, and view the status of Oracle Business Intelligence system components using the OPMN command line:

1. On the Managed Server, go to the following directory that contains the OPMN command line tool:

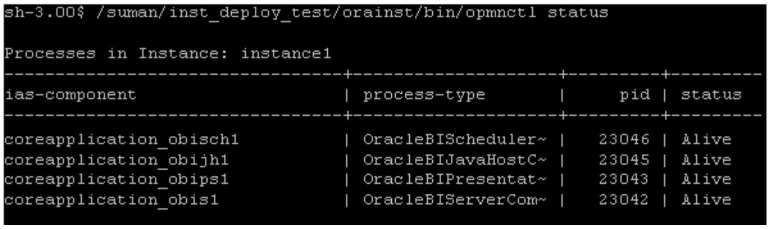
ORACLE_INSTANCE/bin

2. Run the OPMN tool using the appropriate opmnctl command.

   Use the following commands to view status, start, stop, and restart the system components using OPMN on Windows (for UNIX use the command ./opmnctl).

   - `opmnctl status`

     Shows the component names and the status of all system components, as shown in the following figure.



```
sh-3.00$ /suman/inst_deploy_test/orainst/bin/opmnctl status

Processes in Instance: instance1
---------------------------------+--------------------+---------+---------
ias-component                    | process-type       |     pid | status
---------------------------------+--------------------+---------+---------
coreapplication_obisch1          | OracleBIScheduler~ |   23046 | Alive
coreapplication_objh1            | OracleBIJavaHostC~ |   23045 | Alive
coreapplication_obips1           | OracleBIPresentat~ |   23043 | Alive
coreapplication_obis1            | OracleBIServerCom~ |   23042 | Alive
```

   - `opmnctl startall`

     Starts OPMN and all Oracle Business Intelligence system components.

     The bi_server1 (Managed Server) must be running before you can start OPMN. For information, see Section 4.8, "Confirming Whether the Managed Server is Running, and Starting it if Necessary" and Section 4.9, "Starting and Stopping Managed Servers on Windows and UNIX."

   - `opmnctl start`

     Starts OPMN only.

   - `opmnctl startproc ias-component= <component_name>`

     Starts a particular system component. For example, where coreapplication_obips1 is the Oracle BI Presentation Server:

     `opmnctl startproc ias-component=coreapplication_obips1`

     You view system component names in Fusion Middleware Control by displaying the system components Availability region of the Availability tab in the Capacity Management page, and expanding the required entry in the Name column. For more information, see Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components." System component names are displayed in the format coreapplication_<*component_name*> (for example coreapplication_obips1 is the name of the BI Presentation Server component).

   - `opmnctl stopall`

     Stops all managed system components.

   - `opmnctl stopproc ias-component= <component_name>`

     Stops a particular system component.

   - `opmnctl restartproc ias-component= <component_name>`

     Restarts a particular system component.

   - `opmnctl shutdown`

     Stops opmn and all managed system components.

Note: The OPMN configuration file is in the *ORACLE_INSTANCE*/config/OPMN/opmn/opmn.xml directory on UNIX and specifies the exact environment that is used to launch each Oracle Business Intelligence system component. It contains the PATH and LD_LIBRARY_PATH values for the system components and the port number for OPMN and optional SSL configuration.

4.5.2 Using a Windows Service to Start and Stop Oracle Business Intelligence System Components

On Windows, the Oracle Business Intelligence system components are controlled by the Oracle Process Manager (OPMN) Windows Service. To view and manage Windows Services, use one of the following methods:

- Run the following command: `C:\WINDOWS\system32\services.msc`
- From the Start Menu, select **Programs**, then **Administrative Tools** , then **Services**.

The bi_server1 (Managed Server) must be running before you can start OPMN. For information, see Section 4.8, "Confirming Whether the Managed Server is Running, and Starting it if Necessary" and Section 4.9, "Starting and Stopping Managed Servers on Windows and UNIX."

For information about configuring OPMN to start automatically on UNIX, see Section 4.5.3, "Configuring OPMN to Start and Stop Oracle Business Intelligence System Components Automatically on Linux."

To start or restart Oracle Business Intelligence system components using a Windows Service:

1. In the Services window, locate and start the Oracle Process Management (instance_number) option. The Oracle Business Intelligence system components take as long as a minute to start.

   Note: In the Microsoft Windows environment, the Services Manager returns a false error that it cannot start the server. This Microsoft Services Manager issue causes the Services Manager to time out after five minutes. Refreshing the Services window, however, shows that Oracle Business Intelligence is still starting.

   If the Oracle Process Manager fails to start, then errors are logged. Use Fusion Middleware Control to check log file entries and determine why a service has failed. For information, see Chapter 8, "Diagnosing and Resolving Issues in Oracle Business Intelligence."

2. If you run Microsoft IIS, then start the World Wide Web Publishing service.
3. In the ODBC Driver Manager, test that the DSN called AnalyticsWeb_coreapplication (of type Oracle BI Server) connects to the Oracle BI Server.

   Use the user name and password (case-sensitive) that was created during installation for the demonstration repository database.

To stop the Oracle Business Intelligence system components using a Windows Service:

1. In the Services window, locate and stop the Oracle Process Manager service.
2. If you run Microsoft IIS, then also stop the World Wide Web Publishing service.

3. If you run Oracle WebLogic Server, then see Section 4.4, "Using Oracle WebLogic Server Administration Console to Start and Stop Java Components."

4.5.3 Configuring OPMN to Start and Stop Oracle Business Intelligence System Components Automatically on Linux

You can configure OPMN to start Oracle Business Intelligence system components automatically when a Linux operating system starts, and to stop when the operating system shuts down. To configure OPMN to start and stop automatically on Linux, you must include the following commands in the OPMN startup or shutdown scripts respectively:

```
daemon <ORACLE_INSTANCE>/bin ./opmnctl startall
daemon <ORACLE_INSTANCE>/bin ./opmnctl stopall
```

For more information, see the operating system documentation.

4.5.4 Using the Oracle BI Systems Management API to Programmatically Start and Stop Oracle Business Intelligence

You can use the Oracle BI Systems Management API to programmatically start and stop Oracle Business Intelligence.

For information, see Section 30.3.2, "Starting and Stopping Oracle Business Intelligence Using the Oracle BI Systems Management API."

4.6 Confirming Whether the Administration Server is Running

The Administration Server must be running before you can administer Oracle Business Intelligence system and Java components.

You can confirm whether the Administration Server is running by starting a Web browser and entering a URL that should display the Oracle WebLogic Server Administration Console login page.

To confirm whether the Administration Server is running:

1. Enter the following URL into a Web browser:

   http://<*host*>:<*port*>/console/

   For example:

   http://mycomputer:7001/console/

2. Refresh the Web browser window.

3. If the Administration Server is running, then the Oracle WebLogic Server Administration Console login page is displayed.

   If the login page for Oracle WebLogic Server Administration Console is not displayed, then the Administration Server is not running, and you must start it.

   For information, see Section 4.7, "Starting and Stopping the Administration Server and all Oracle Business Intelligence Java Components on Windows and UNIX."

4.7 Starting and Stopping the Administration Server and all Oracle Business Intelligence Java Components on Windows and UNIX

You can start or stop the Administration Server and all Oracle Business Intelligence Java components on Windows and UNIX.

For information, see Section 4.6, "Confirming Whether the Administration Server is Running."

To start the Administration Server and all Oracle Business Intelligence Java components on Windows and UNIX:

If the Administration Server is not running, then start it using the following information.

Use the appropriate method for the operating system:

Before you can start the Administration Server, the database that you specified during installation must be running, or JDBC errors prevent startup.

- On Windows, use the following method:
  - Open an MS-DOS window and change the directory to *MW_HOME*\user_projects\domains\bifoundation_domain\bin. Then, run the following command (which requires the administrator user name and password):

    ```
    startWebLogic.cmd -start
    ```

    You can also stop the Java components from the MS-DOS window in which they were started, if you press Ctrl+C.
- On UNIX, open a shell prompt and change the directory to *MW_HOME*/user_projects/domains/bifoundation_domain/bin. Then, run the following command (which requires the administrator user name and password):

  ```
  ./startWebLogic.sh
  ```

> **Caution:** If you start the Administration Server from a Windows or UNIX command line window, then do not close the window later on, or the server terminates (unless the server is started as a background process from the command line).

To stop the Administration Server and Oracle Business Intelligence Java components on Windows and UNIX:

Use the appropriate method for the operating system:

- On Windows, use one of the following methods:
  - Display the Start Menu, click **All Programs**, **Oracle WebLogic**, **User Projects**, and **bifoundation_domain**.

    Select the **Stop Admin Server for WebLogic Server Domain** menu option

    An MS-DOS window indicates the progress of the processing steps of stopping the Administration Server.
  - Open an MS-DOS window and change the directory to *MW_HOME*\user_projects\domains\bifoundation_domain\bin. Then, run the following command (which requires the administrator user name and password):

    ```
    stopWebLogic.cmd -stop
    ```

You can also stop the Java components from the MS-DOS window in which they were started, if you press Ctrl+C.

- On UNIX, open a shell prompt and change the directory to *MW_HOME*/user_projects/domains/bifoundation_domain/bin. Then, run the following command (which requires the administrator user name and password):

  ```
  ./stopWebLogic.sh
  ```

  You can also use the process termination command for the operating system in use (for example, kill on UNIX). Java indicates on the console window that it is shutting down when it receives a shutdown signal.

4.8 Confirming Whether the Managed Server is Running, and Starting it if Necessary

You can use Oracle WebLogic Server Administration Console or Fusion Middleware Control to confirm whether the Managed Server is running, and start it if necessary.

To confirm whether the Managed Server is running, and start it if necessary using Oracle WebLogic Server Administration Console:

1. Log in to Oracle WebLogic Server Administration Console using the following URL:

   http://<*host*>:7001/console

   For information, see Section 2.3, "Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console."

2. Under the bifoundation_domain, expand the Environment node in the navigation tree, and click **Servers**.

3. Display the Configuration tab.

   There should be two servers listed: AdminServer (the Administration Server) and bi_server1 (Managed Server).

4. Start the Managed Server, if not running:

   a. Display the Control tab.

   b. Select the box next to the Managed Server (for example, bi_server1).

   c. Click **Start**.

      Repeat this step for each Managed Server.

   For information about starting Managed Servers on Windows and UNIX, see Section 4.9, "Starting and Stopping Managed Servers on Windows and UNIX."

5. (Optional) Perform an OPMN stopall for the system components, and then a startall. For information, see Section 4.5.1, "Using the OPMN Command Line to Start, Stop, Restart, and View the Status of Oracle Business Intelligence System Components."

To confirm whether a Managed Server is running, and start it if necessary using Fusion Middleware Control:

1. Log in to Fusion Middleware Control using the following URL:

   http://<*host*>:7001/em

For information, see Section 2.2, "Centrally Managing Oracle Business Intelligence Components Using Fusion Middleware Control."

2. Under the WebLogic Domain expand the bifoundation_domain node in the navigation tree, and select bi_cluster.

   Expand the bi_cluster node to display the Managed Servers (for example, bi_server1). Fusion Middleware Control displays the status in the Deployments region.

3. Select bi_cluster, and click the WebLogic Cluster menu.
4. Select the **Control** option from the menu, and select **Start Up** to start the Managed Servers.
5. (Optional) Perform an OPMN stopall for the system components, and then a startall. For information, see Section 4.5.1, "Using the OPMN Command Line to Start, Stop, Restart, and View the Status of Oracle Business Intelligence System Components."

4.9 Starting and Stopping Managed Servers on Windows and UNIX

You can start and stop Managed Servers on Windows and UNIX.

To start Managed Servers on Windows and UNIX:

Use the appropriate method for the operating system:

- Open an MS-DOS window and change the directory to *MW_HOME*\user_projects\domains\bifoundation_domain\bin. Then, run the following command (which requires the administrator user name and password), for example:

  ```
  startManagedWebLogic.cmd bi_server1 http://<administration_server_host_
  name>:7001
  ```

  You can also stop the Java components from the MS-DOS window in which they were started, if you press Ctrl+C.

- On UNIX, open a shell prompt and change the directory to *MW_HOME*/user_projects/domains/bifoundation_domain/bin. Then, run the following command (which requires the administrator user name and password), for example:

  ```
  ./startManagedWebLogic.sh bi_server1 http://<administration_server_host_
  name>:7001
  ```

Caution: If you start Managed Servers from a Windows or UNIX command line window, then do not close the window later on, or the server terminates.

Do not use the `startManagedWebLogic.sh` script that is located in *MW_HOME*/wlserver_10.3/common/bin/ because it is not supported.

To stop Managed Servers on Windows and UNIX:

Use the appropriate method for the operating system:

- On Windows, open an MS-DOS window and change the directory to *MW_HOME*\user_projects\domains\bifoundation_domain\bin. Then, run the following command (which requires the administrator user name and password), for example:

```
stopManagedWebLogic.cmd bi_server1 http://<administration_server_host_
name>:7001
```

You can also stop the Java components from the MS-DOS window in which they were started, if you press Ctrl+C.

- On UNIX, open a shell prompt and change the directory to *MW_HOME*/user_projects/domains/bifoundation_domain/bin. Then, run the following command (which requires the administrator user name and password), for example:

  ```
  ./stopManagedWebLogic.sh bi_server1 http://<administration_server_host_
  name>:7001
  ```

  You can also use the process termination command for the operating system in use (for example, kill on UNIX). Java indicates on the console window that it is shutting down when it receives a shutdown signal.

4.10 Troubleshooting System Startup

This section contains solutions that are related to system startup:

- Section 4.10.1, "The Administration Server Fails to Start When the Database is Not Running"
- Section 4.10.2, "The Managed Server is Down"
- Section 4.10.3, "The Oracle BI Server Fails to Start"
- Section 4.10.4, "Oracle BI Presentation Services Fails to Start"
- Section 4.10.5, "Unable to Log On"

4.10.1 The Administration Server Fails to Start When the Database is Not Running

When you start the Administration Server, the repository database that was specified during installation must be running, or else you see JDBC errors that prevent startup.

Problem: The Administration Server fails to start.

If the Administration Server fails to start, then:

- View the Administration Server and Managed Server log files in the following directory:

  \user_projects\domains\bifoundation_domain\servers\AdminServer\logs

  You can also check the Managed Server log files in the following directory:

  \user_projects\domains\bifoundation_domain\servers\bi_server1\logs

Cause: Database Down: in AdminServer.log, "Caused By: java.net.UnknownHostException: yourcomputername" deep in the trace from:

####<Jan 19, 2010 8:04:09 PM PST> <Info> <JDBC> <username> <AdminServer> <[ACTIVE] ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'> <<anonymous>> <Stack trace associated with message 001129 follows:

java.sql.SQLException: The Network Adapter could not establish the connection.

Resolution: Start the database.

4.10.2 The Managed Server is Down

If the Managed Server is down, then use the Oracle WebLogic Server Administration Console, or the command line to start it.

For information, see Section 4.8, "Confirming Whether the Managed Server is Running, and Starting it if Necessary."

4.10.3 The Oracle BI Server Fails to Start

If the BI Server fails to start, then view the log files in the following directory:

<ORACLE_INSTANCE>\diagnostics\logs\OracleBIServerComponent\coreapplication_obis1, or use the log viewer.

4.10.4 Oracle BI Presentation Services Fails to Start

If Presentation Services fails to start, then view the log files in the following directory:

<ORACLE_INSTANCE>\diagnostics\logs\OracleBIPresentationServicesComponent\coreapplication_obips1, or use the log viewer.

If you experience either of the following problems related to Presentation Services failing to start, then refer to the appropriate causes and resolutions:

- **Problem**: Could not create a system user connection to the BI Server during startup.

  Cause: You started Presentation Services before the BI Server.

  Resolution: Use the OPMN stopall command to stop the Business Intelligence system, and then use the startall command. For information, see Section 4.5.1, "Using the OPMN Command Line to Start, Stop, Restart, and View the Status of Oracle Business Intelligence System Components."

- **Problem**: Authentication failed for BISystemUser: invalid user name and password.

  Cause: The Managed Server is not running.

  Resolution: See Section 4.10.2, "The Managed Server is Down."

4.10.5 Unable to Log On

- **Problem**: Cannot log in to Oracle WebLogic Server Administration Console.

  Cause: The Administration Server is not running.

  Resolution: Check to see if http://*<host>*:7001/console starts. If not, start the Administration Server. For information, see Section 4.7, "Starting and Stopping the Administration Server and all Oracle Business Intelligence Java Components on Windows and UNIX."

Part III

Scaling and Deploying for High Availability and Performance

This part explains how to manage deployment, availability, and capacity for Oracle Business Intelligence. It contains the following chapters:

5 Scaling Your Deployment

This chapter describes how to manage the capacity of your deployment. By default, the Oracle Business Intelligence components are installed in a cluster configuration and are scalable. User Web requests can be directed to one of many Oracle BI Presentation Services components. In turn, each Presentation Services component can take advantage of the availability of multiple Oracle BI Servers.

You can expand or reduce the capacity of the system by adjusting the number of processes available to the cluster. Increasing or decreasing the capacity of a system by making effective use of resources is known as *scalability*. A scalable system can handle increasing numbers of requests without adversely affecting response time and throughput.

This chapter contains the following topics:

- Section 5.1, "About Scaling Oracle Business Intelligence"
- Section 5.2, "Vertically Scaling Oracle Business Intelligence"
- Section 5.3, "Horizontally Scaling Oracle Business Intelligence"
- Section 5.5, "Using Fusion Middleware Control to Scale System Components"
- Section 5.4, "Setting Up Shared Files and Directories"
- Section 5.6, "Validating That Your System Has Been Scaled Correctly"

5.1 About Scaling Oracle Business Intelligence

Scaling is the process of increasing or decreasing the capacity of the system by changing the number of processes available to service requests from Oracle Business Intelligence clients. Scaling out a system provides additional capacity, while scaling in a system reduces capacity. Scaling is also a critical part of configuring a deployment for high availability.

Scaling the Oracle Business Intelligence environment applies principally to resource-intensive system processes and Java components. When you deploy more processes, Oracle Business Intelligence can handle more requests while staying responsive to requests.

Vertical scaling involves adding more Oracle Business Intelligence components to the same computer, to make increased use of the hardware resources on that computer. For example, Oracle Business Intelligence can be vertically scaled by increasing the number of system components servicing requests on a given computer and results in increased use of the hardware resources on a given computer.

Horizontal scaling involves adding more computers to the environment. For example, Oracle Business Intelligence is horizontally scaled by distributing the processing of requests across multiple computers.

You can scale both Oracle Business Intelligence Java components and system components. See Section 1.4, "About Oracle BI Java Components and System Components" for more information about these components.

The three system components that support both horizontal and vertical scale-out are Oracle BI Presentation Services, the Oracle BI Server, and the JavaHost.

Oracle BI Scheduler uses Presentation Services and Oracle BI Server processes to perform computationally intense work on its behalf, while the Cluster Controller only manages other components and does not itself do any computationally intense work. Because of this, there is no need to scale out either Oracle BI Scheduler or the Cluster Controller. You can distribute these two processes as needed for high availability deployments, but they do not need to be scaled for capacity.

5.1.1 How Do I Know When to Scale Out Processes?

Scale out system components and Managed Servers based on observed load. You can use the performance metrics that are provided in Fusion Middleware Control to monitor process state and to determine when you must increase capacity to improve performance. For example, you might want to add an additional computer to the deployment when CPU usage is over 50%, or when memory use is close to the system limit. See Section 7.1, "Monitoring Service Levels" for more information about viewing system metrics.

You also must scale out processes to achieve redundancy when you want to configure a highly available Oracle Business Intelligence environment. See Section 6, "Deploying Oracle Business Intelligence for High Availability" for more information.

5.1.2 What Processes Should I Scale?

Oracle Business Intelligence provides support for scale-out using a combination of the Oracle Business Intelligence installer (for horizontal scale-out) and Fusion Middleware Control (to scale system components both vertically and horizontally).

Follow these guidelines for scaling Managed Servers and system components:

- Make sure to run one Managed Server on each computer in the deployment. The Oracle Business Intelligence installer automatically provisions one Managed Server. Do not disable or remove it.
- It is not necessary to run multiple Managed Servers on a given computer.
- Do not remove individual Java components, because many perform essential services for the system. Keep a full set of Java components on each Managed Server. Any unused components should not have a significant performance impact.
- You can decide based on observed load which system components to run on each computer. You can have 0 or more of each component type on a given computer in the deployment. For example, you can have three Oracle BI Servers, two JavaHosts, and four Presentation Services components.
- You do not need to scale any configured HTTP servers along with either the Managed Servers or system components. HTTP server configuration is independent of the number of processes that you run.

5.2 Vertically Scaling Oracle Business Intelligence

When you have multiple instances of a given Oracle Business Intelligence component in the deployment, you should first configure shared files and directories for the clustered components to use. Then, use Fusion Middleware Control to increase the number of system components that are running on the existing Oracle Business Intelligence host.

It is not necessary to run multiple Managed Servers on a given computer. Because of this, vertical scale-out only applies to Oracle Business Intelligence system components.

Table 5–1 lists the tasks that you must perform to vertically scale Oracle Business Intelligence.

Table 5–1 Task Summary for Vertical Scale-Out

Task	Where to Go for More Information
Configure shared files and directories for the Oracle BI repository, Oracle BI Presentation Catalog, global cache, and Oracle BI Scheduler scripts	Section 5.4, "Setting Up Shared Files and Directories"
Scale out the necessary Oracle Business Intelligence system components	Section 5.5, "Using Fusion Middleware Control to Scale System Components"

5.3 Horizontally Scaling Oracle Business Intelligence

As with vertical scale-out, when you have multiple instances of a given Oracle Business Intelligence component in the deployment, you should first configure shared files and directories for the clustered components to use.

Next, to horizontally scale out the Oracle Business Intelligence Java and system components, install Oracle Business Intelligence on the new host by running the Oracle Business Intelligence installer and selecting **Enterprise Install**. Then, select the **Scale Out BI System** option.

When the installation completes, the Oracle Business Intelligence Java components are available on the new Managed Server. Note that the first Managed Server for Oracle Business Intelligence Java components is called bi_server1. When you scale out using the Enterprise Install with Scale Out BI System option, additional Managed Servers are called bi_server2, bi_server3, and so on.

After you complete the Enterprise Install, you must provision the scale out for the Oracle Business Intelligence system components on the new host using Fusion Middleware Control. You also typically configure an HTTP server and load balancer to distribute requests across Managed Servers.

Table 5–2 lists the tasks that you must perform to horizontally scale Oracle Business Intelligence.

Table 5–2 Task Summary for Horizontal Scale-Out

Task	Where to Go for More Information
Configure shared files and directories for the Oracle BI repository, Oracle BI Presentation Catalog, global cache, and Oracle BI Scheduler scripts	Section 5.4, "Setting Up Shared Files and Directories"
Run the Oracle Business Intelligence installer on the new host and choose the Enterprise Install option	"Enterprise Install to Scale Out Existing Installations" in *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*

Table 5–2 (Cont.) Task Summary for Horizontal Scale-Out

Task	Where to Go for More Information
Scale out the Oracle Business Intelligence system components on the new host	Section 5.5, "Using Fusion Middleware Control to Scale System Components"
Configure an HTTP server with a load balancer to distribute requests across multiple Managed Servers	*Oracle Fusion Middleware Using Web Server Plug-Ins with Oracle WebLogic Server* "Load Balancing in a Cluster" in *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*

5.4 Setting Up Shared Files and Directories

When you have multiple instances of a given Oracle Business Intelligence component, you can share certain files and directories on a shared storage device such as NAS or SAN to simplify management of your system. It is recommended that you host the Oracle BI repository (RPD file), Oracle BI Presentation Catalog, global cache, and shared Oracle BI Scheduler scripts on shared storage before you scale out Oracle Business Intelligence components.

This section contains the following topics:

- Section 5.4.1, "Uploading and Sharing the Oracle BI Repository"
- Section 5.4.2, "Sharing the Oracle BI Presentation Catalog"
- Section 5.4.3, "Setting Up the Global Cache"
- Section 5.4.4, "Setting the Scheduler Script Path and Default Script Path"

5.4.1 Uploading and Sharing the Oracle BI Repository

It is recommended that you configure a repository publishing directory so that the repository can be shared by all Oracle BI Servers participating in a cluster. This directory holds the master copies of repositories that are edited in online mode. The clustered Oracle BI Servers examine this directory upon startup for any repository changes.

To share the Oracle BI repository:

1. Create a shared directory for the Repository Publishing Directory. The Master BI Server must have read and write access to this directory. All other Oracle BI Servers must have read access.

2. Use the Repository tab of the Deployment page in Fusion Middleware Control to designate the shared location for the Oracle BI repository. See Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location" for more information.

   Note that when you configure a shared repository, the repository file that you upload in Fusion Middleware Control is copied to both the shared location and to the local repository directory for each Oracle BI Server. Do not delete the local repository copies, or set read-only file permissions on them, because they are needed by the system.

5.4.2 Sharing the Oracle BI Presentation Catalog

It is recommended that you configure a shared directory for the Oracle BI Presentation Catalog so that it can be shared by all Presentation Services components in a cluster.

Because the Oracle BI Presentation Catalog consists of a large number of heavily accessed small files, there are two important considerations for the shared file system:

- **File Limits:** The Oracle BI Presentation Catalog can consist of thousands of files. In many cases, this might exceed file limits for shared file systems. Check the storage vendor documentation for instructions on extending the file limit.
- **Snapshots:** Backup activity such as snapshots might adversely affect the performance of Oracle BI Presentation Catalog files, which are small, dynamic files. Ensure that snapshot activity is at a reasonable level that maximizes performance without impacting availability.

To share the Oracle BI Presentation Catalog:

1. Create a network share for the Oracle BI Presentation Catalog. All Presentation Services components in the cluster must have read and write access to this share.
2. Place the catalog on the network share.
3. Use the Repository tab of the Deployment page in Fusion Middleware Control to change the location of the Oracle BI Presentation Catalog to the shared location. See Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location" for more information.

5.4.3 Setting Up the Global Cache

The global cache is a query cache that is shared by all Oracle BI Servers participating in a cluster. For more information, see Section 7.4.6, "About the Global Cache."

It is recommended that you configure the global cache so that cache seeding and purging events can be shared by all Oracle BI Servers participating in a cluster.

To set up the global cache:

1. Create a shared directory for the global cache. All Oracle BI Servers must have read and write access to this directory.
2. Use the Performance tab of the Capacity Management page in Fusion Middleware Control to set the **Global cache path** and **Global cache size** options. See Section 7.5.4, "Using Fusion Middleware Control to Set Global Cache Parameters" for more information.

5.4.4 Setting the Scheduler Script Path and Default Script Path

If you use server-side scripts with Oracle BI Scheduler, it is recommended that you configure a shared directory for the scripts so that they can be shared by all Oracle BI Scheduler components in a cluster.

In this release, the Action Framework supersedes the need to use server-side scripts for agents. See "Working with Actions" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for more information about using Actions.

Perform these steps only if you are use server-side scripts from a previous release.

To share Oracle BI Scheduler scripts:

1. Create network shares for the Oracle BI Scheduler scripts. The Oracle BI Scheduler servers must have read and write access to this share.
2. Copy the default and custom Oracle BI Scheduler scripts to the corresponding network shares created for the Oracle BI Scheduler scripts.

3. Update the SchedulerScriptPath and DefaultScriptPath elements of the Oracle BI Scheduler instanceconfig.xml file.

   You must update this file for each Oracle BI Scheduler component in the deployment. See Section 20.3.3.1, "General Scheduler Configuration Settings That Affect Agents" for more information about setting these two parameters.

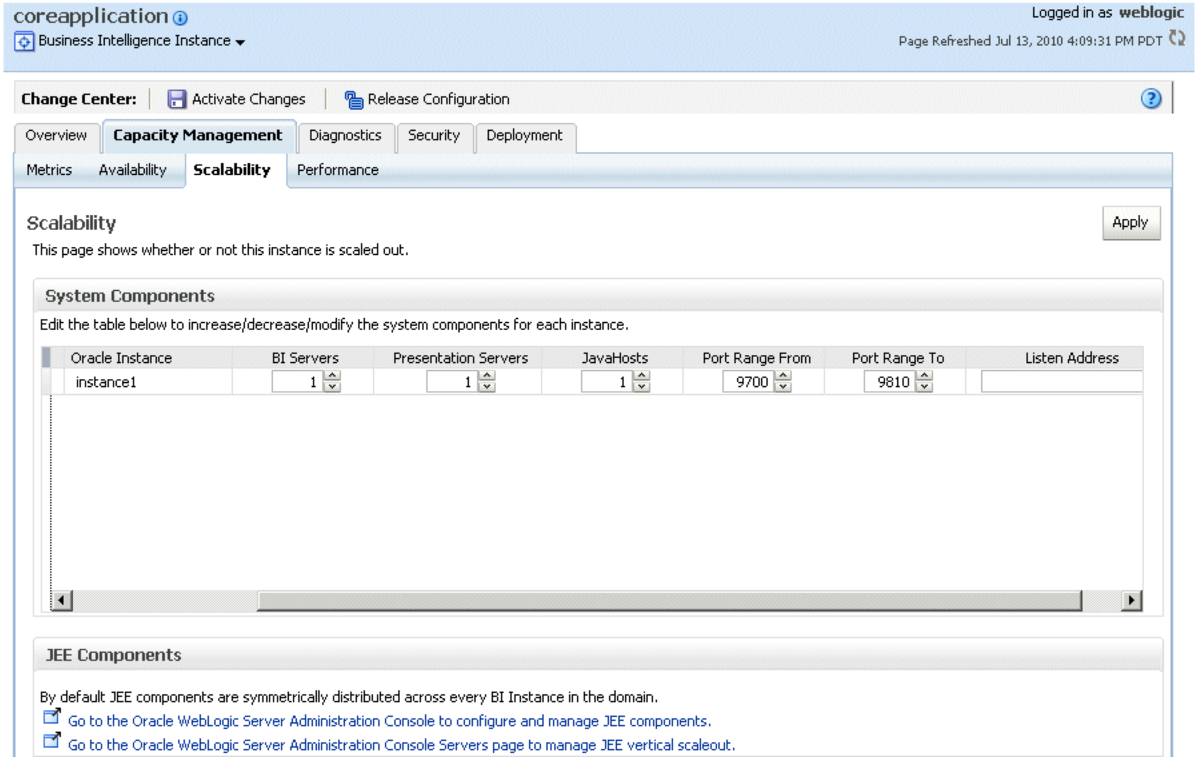
5.5 Using Fusion Middleware Control to Scale System Components

You can use the Scalability tab of the Capacity Management page in Fusion Middleware Control to scale in or scale out the Oracle BI Server, Presentation Services, or JavaHost system components.

You can follow the procedure described in this section to scale the system components both vertically (on the same computer) or horizontally (on multiple computers).

Figure 5–1 shows the Scalability tab of the Capacity Management page.

Figure 5–1 Scalability Tab of Capacity Management Page in Fusion Middleware Control



Scalability

This page shows whether or not this instance is scaled out.

System Components

Edit the table below to increase/decrease/modify the system components for each instance.

Oracle Instance	BI Servers	Presentation Servers	JavaHosts	Port Range From	Port Range To	Listen Address
instance1	1	1	1	9700	9810	

JEE Components

By default JEE components are symmetrically distributed across every BI Instance in the domain.

Go to the Oracle WebLogic Server Administration Console to configure and manage JEE components.

Go to the Oracle WebLogic Server Administration Console Servers page to manage JEE vertical scaleout.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To scale Oracle Business Intelligence system components:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."

2. Display the Scalability tab of the Capacity Management page.

   Click the **Help** button on the page to access the page-level help for its elements.

3. Click **Lock and Edit Configuration** to allow changes to be made.
4. Change the number of **BI Servers**, **Presentation Servers**, or **JavaHosts** using the arrow keys.

   If you have scaled out the Oracle BI System on a new host using the Enterprise Install option or the Configuration Assistant, then the number of system components configured on that host is zero. Add system components on the new host to complete the horizontal scale-out operation.

   To vertically scale the system components, increment the number of each component to be greater than 0 for a given host.
5. A default port range is assigned automatically for use by the components within the given Oracle instance. Note that Oracle BI Scheduler and Cluster Controller component processes in this instance are also assigned ports within this range, in addition to the BI Server, Presentation Services, and JavaHost component processes.

   Typically, you can keep the default port range. If necessary, enter a different range of ports available using the **Port Range From** and **Port Range To** arrows, or enter a value directly.
6. Optionally, for **Listen Address**, you can enter the DNS name or IP address for the components in the given Oracle instance to use to listen for incoming connections. Typically, you can keep this element blank (the default value) to indicate that the components should listen on *all* available local interfaces.
7. Click **Apply**, then click **Activate Changes**.
8. Return to the Business Intelligence Overview page and click **Restart**.

   The changes are automatically written to the corresponding configuration files, and the new processes join the cluster.

For information on using methods in the Oracle BI Systems Management API to scale out components, see Chapter 30, "Introducing the Oracle BI Systems Management API."

5.6 Validating That Your System Has Been Scaled Correctly

You can use Fusion Middleware Control and the Oracle WebLogic Server Administration Console to verify the status of the scaled-out components.

This section contains the following topics:

- Section 5.6.1, "Using Fusion Middleware Control to View System Component Availability"
- Section 5.6.2, "Using the Administration Console to View Managed Server Availability"

5.6.1 Using Fusion Middleware Control to View System Component Availability

You can use Fusion Middleware Control to view the status of all system components in your deployment.

To view status for system components:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."

2. Display the Availability tab of the Capacity Management page.

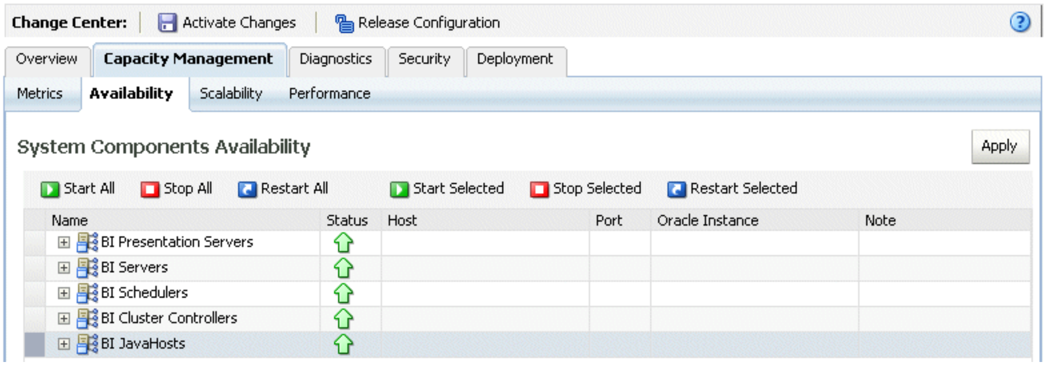
   On this page, you can:

   - View the status of all configured system components
   - View the host, Oracle instance, and port where the each system component is currently running
   - Start, stop, or restart individual system components
   - Start, stop, or restart all system components of a particular type
   - Start, stop, or restart all processes

   Click the **Help** button on the page to access the page-level help for its elements.

Figure 5–2 shows the Availability tab of the Capacity Management page, with status information displayed in the System Components Availability table.

Figure 5–2 Availability Tab of Capacity Management Page in Fusion Middleware Control



Name	Status	Host	Port	Oracle Instance	Note
⊞ BI Presentation Servers	⇧				
⊞ BI Servers	⇧				
⊞ BI Schedulers	⇧				
⊞ BI Cluster Controllers	⇧				
⊞ BI JavaHosts	⇧				

5.6.2 Using the Administration Console to View Managed Server Availability

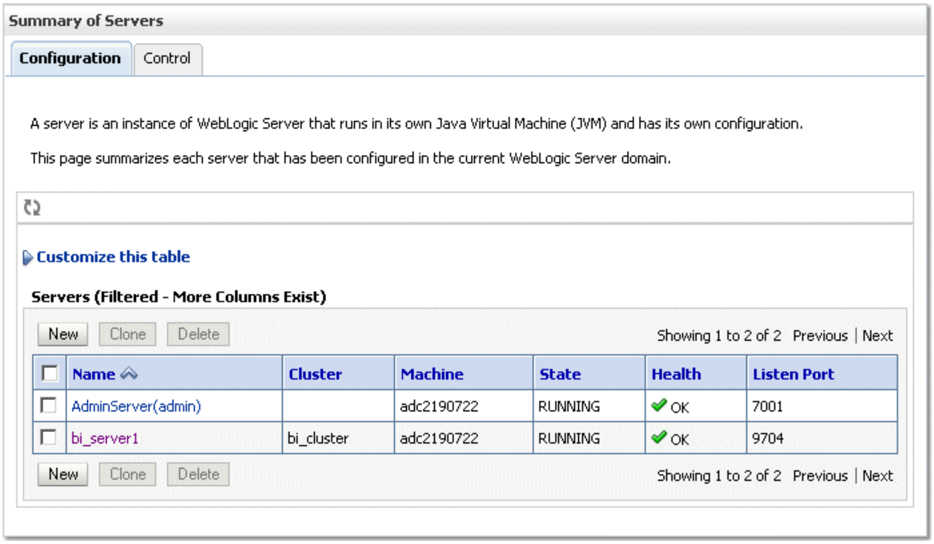
You can use the Administration Console to view the status of all Managed Servers in your deployment.

To view status for Managed Servers:

1. Log in to the Oracle WebLogic Server Administration Console.
2. Select **Environment**, then select **Servers** to go to the Summary of Servers page. On this page, you can see any Managed Servers that were added on new hosts in your deployment.

Figure 5–3 shows the Summary of Servers page.

Figure 5–3 Summary of Servers Page in Oracle WebLogic Server Administration Console



A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.

This page summarizes each server that has been configured in the current WebLogic Server domain.

Servers (Filtered - More Columns Exist)

□	Name	Cluster	Machine	State	Health	Listen Port
□	AdminServer(admin)		adc2190722	RUNNING	✓ OK	7001
□	bi_server1	bi_cluster	adc2190722	RUNNING	✓ OK	9704

6 Deploying Oracle Business Intelligence for High Availability

This chapter provides information about how to configure Oracle Business Intelligence components for high availability. It also describes the functionality available in Fusion Middleware Control to manage system availability, and provides information about using the Cluster Manager in the Administration Tool.

This chapter does not provide information about setting up additional high availability configuration for other components in the stack, including database tier, Web tier, Administration Server, and identity management availability. For more information about these topics and how they relate to Oracle Business Intelligence deployments, see the following documents:

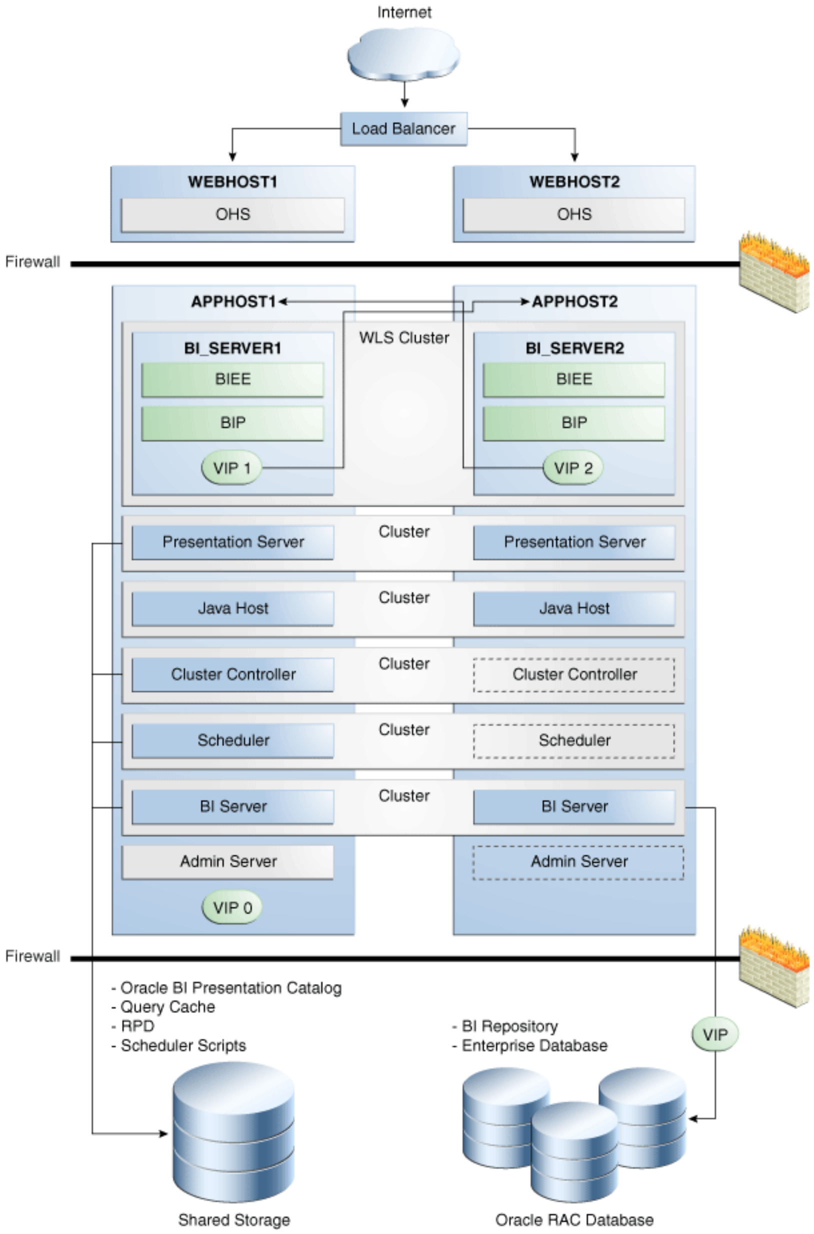
- "Configuring Oracle Business Intelligence for High Availability" in *Oracle Fusion Middleware High Availability Guide* explains how to implement high availability across the stack, including how to configure a fault tolerant HTTP load balancer and a highly available database for the Oracle Business Intelligence schemas
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence* explains how to deploy Oracle Business Intelligence based on an architectural blueprint that follows Oracle recommended best practices for security and high availability, including Web tier, database tier, Administration Server, and identity management availability

This chapter contains the following sections:

6.1 About Oracle Business Intelligence Components in a Clustered Environment

Figure 6–1 shows the system components and Java components in a highly available Oracle Business Intelligence deployment. See Section 1.4, "About Oracle BI Java Components and System Components" for more information about system components and Java components.

Figure 6–1 A Highly Available Oracle Business Intelligence Deployment



In Figure 6–1, the Oracle Business Intelligence Java components are deployed on the BI_SERVER1 and BI_SERVER2 Managed Servers on APPHOST1 and APPHOST2. These Managed Servers are configured in an Oracle WebLogic cluster.

Oracle BI Presentation Services, JavaHost, Oracle BI Cluster Controller, Oracle BI Scheduler, and Oracle BI Server are system components installed on APPHOST1 and

APPHOST2 and configured as a cluster. The Cluster Controller and Oracle BI Scheduler on APPHOST2 are passive (they are started but do not service requests) and are only made active if APPHOST1 components fail.

In the data tier, shared external storage is configured to store the Oracle BI Presentation Catalog, Oracle BI Server global cache, Oracle BI repository, and Oracle BI Scheduler script data.

6.1.1 Recommendations for Availability

In a production system, it is recommended that you deploy two or more instances of every component on two or more computers, so that each component type has an instance running on more than one computer for fault tolerance. This configuration provides redundancy for Managed Servers and system components, an essential requirement for high availability and failover. You can see whether the system has any single points of failure by using the Availability tab of the Capacity Management page in Fusion Middleware Control. See Section 6.1.2, "Using Fusion Middleware Control to Identify Single Points of Failure" for more information.

You can also ensure high availability by configuring redundancy in the database tier (Oracle RAC recommended), Web tier, and for the Administration Server. See "Configuring Oracle Business Intelligence for High Availability" in *Oracle Fusion Middleware High Availability Guide* for more information.

Note also the following requirements:

- All Oracle BI Servers participating in the cluster must be within the same domain and on the same LAN subnet. Geographically separated computers are not supported.
- The clock on each server participating in a cluster must be kept in synchronization. Out-of-sync clocks can skew reporting.

6.1.2 Using Fusion Middleware Control to Identify Single Points of Failure

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To identify single points of failure:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Availability tab of the Capacity Management page.

   On this page, you can view recommendations about whether to scale out system components or configure primary/secondary system components.

   Click the **Help** button on the page to access the page-level help for its elements.
3. If you must scale out the Oracle BI Server, Oracle BI JavaHost, or Oracle BI Presentation Services, then you can click **Scale Out Selected** in the Single Points of Failure section to go to the Scalability tab to scale out a system component. See Section 5.5, "Using Fusion Middleware Control to Scale System Components" for more information.
4. If you have a Cluster Controller or Oracle BI Scheduler that must be configured, the Single Points of Failure table displays the message "Configure

Primary/Secondary." See Section 6.2.1, "Using Fusion Middleware Control to Configure Primary and Secondary Instances" for information about how to do this.

6.2 Configuring Oracle Business Intelligence Components for High Availability

To configure Oracle Business Intelligence for high availability, you must ensure that the system has no single points of failure by scaling out the Oracle BI Server, Presentation Services, and the JavaHost so that you have at least two of each component type, distributed across at least two computers.

You also must configure primary and secondary instances of the Cluster Controller and Oracle BI Scheduler, so that the primary and secondary instances for each component type are distributed across two different computers.

Table 6–1 lists the tasks you must perform to configure high availability for Oracle Business Intelligence.

Table 6–1 Task Summary for Configuring High Availability

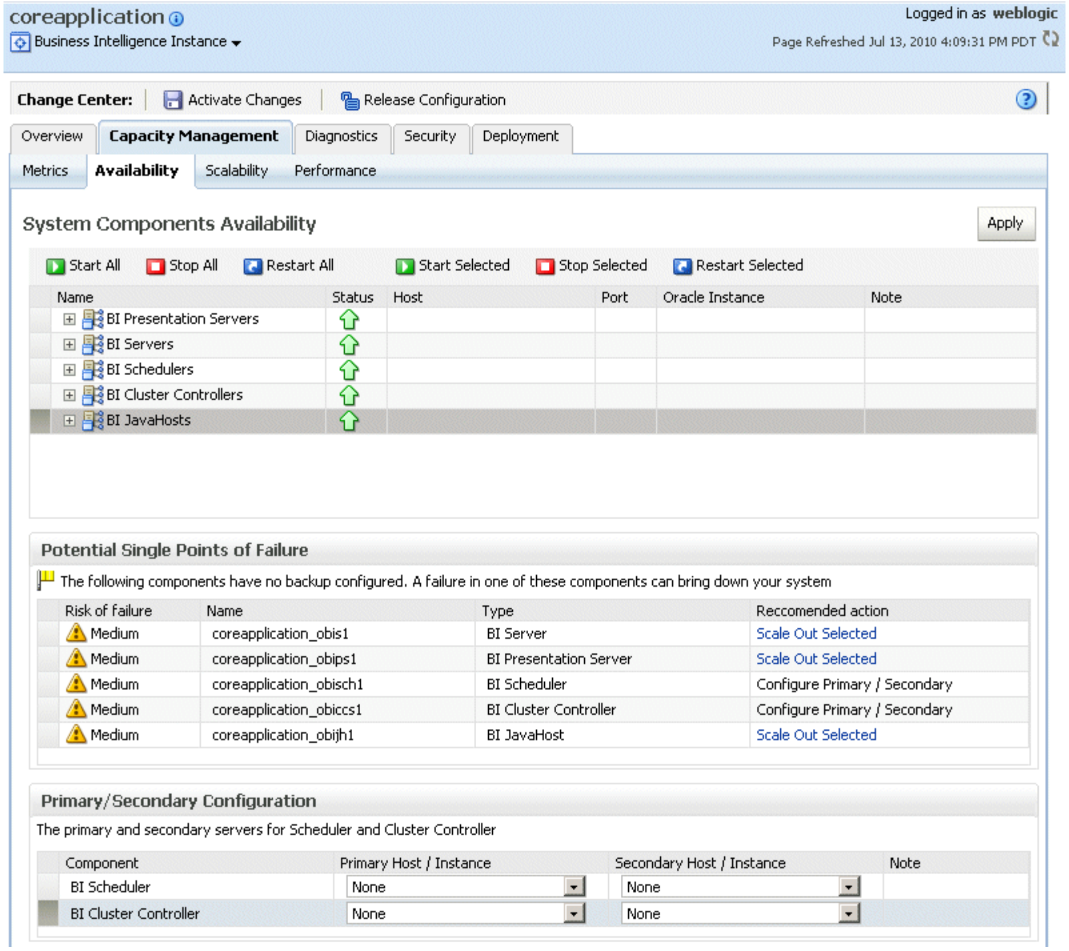
Task	Where to Go for More Information
Horizontally scale out the Oracle Business Intelligence deployment so that it includes two computers with a full set of Java and system components on each host. This task includes running the Oracle Business Intelligence installer, configuring shared files and directories, and scaling out system components using Fusion Middleware Control.	Section 5.3, "Horizontally Scaling Oracle Business Intelligence"
Configure primary and secondary instances of the Cluster Controller and Oracle BI Scheduler.	Section 6.2.1, "Using Fusion Middleware Control to Configure Primary and Secondary Instances"
Verify that the new components are available.	Section 5.6.1, "Using Fusion Middleware Control to View System Component Availability"

6.2.1 Using Fusion Middleware Control to Configure Primary and Secondary Instances

You can use Fusion Middleware Control to configure primary and secondary instances of the Cluster Controller and Oracle BI Scheduler.

Figure 6–2 shows the Availability tab of the Capacity Management page.

Figure 6–2 Availability Tab of Capacity Management Page in Fusion Middleware Control



coreapplication

Business Intelligence Instance

Logged in as **weblogic**

Page Refreshed Jul 13, 2010 4:09:31 PM PDT

Change Center: Activate Changes | Release Configuration

Overview | **Capacity Management** | Diagnostics | Security | Deployment

Metrics | **Availability** | Scalability | Performance

System Components Availability

Apply

Start All | Stop All | Restart All | Start Selected | Stop Selected | Restart Selected

Name	Status	Host	Port	Oracle Instance	Note
BI Presentation Servers	⇧				
BI Servers	⇧				
BI Schedulers	⇧				
BI Cluster Controllers	⇧				
BI JavaHosts	⇧				

Potential Single Points of Failure

The following components have no backup configured. A failure in one of these components can bring down your system

Risk of failure	Name	Type	Reccomended action
⚠ Medium	coreapplication_obis1	BI Server	Scale Out Selected
⚠ Medium	coreapplication_obips1	BI Presentation Server	Scale Out Selected
⚠ Medium	coreapplication_obisch1	BI Scheduler	Configure Primary / Secondary
⚠ Medium	coreapplication_obiccs1	BI Cluster Controller	Configure Primary / Secondary
⚠ Medium	coreapplication_obijh1	BI JavaHost	Scale Out Selected

Primary/Secondary Configuration

The primary and secondary servers for Scheduler and Cluster Controller

Component	Primary Host / Instance	Secondary Host / Instance	Note
BI Scheduler	None	None	
BI Cluster Controller	None	None	

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To configure primary and secondary instances of the Cluster Controller and Oracle BI Scheduler:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Availability tab of the Capacity Management page.

   On this page, you can configure primary and secondary instances of the Cluster Controller and Oracle BI Scheduler.

   Click the **Help** button on the page to access the page-level help for its elements.
3. Click **Lock and Edit Configuration** to allow changes to be made.
4. In the **Primary/Secondary Configuration** section, select the host and Oracle instance on which you want to run the primary or secondary Cluster Controller or Oracle BI Scheduler.
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to manage availability, see Chapter 30, "Introducing the Oracle BI Systems Management API."

6.3 How Do Oracle Business Intelligence Components Provide Availability?

This section provides information about how the Oracle Business Intelligence components interact with each other in a clustered environment, and explains how Oracle Business Intelligence ensures availability for each component type.

- Section 6.3.1, "About the Cluster Controller"
- Section 6.3.2, "About Clustered Oracle BI Servers and the Master BI Server"
- Section 6.3.3, "About Clustered Oracle BI Scheduler Instances"
- Section 6.3.4, "About the Oracle BI Cluster Server"
- Section 6.3.5, "About Clustered Presentation Services Instances"
- Section 6.3.6, "About Clustered JavaHost Instances"
- Section 6.3.7, "About Web Tier Components"

6.3.1 About the Cluster Controller

The Cluster Controller serves as the first point of contact for new requests from Oracle BI Presentation Services and other clients of the Oracle BI Server. The Cluster Controller determines which Oracle BI Server in the cluster should receive incoming requests, based on Oracle BI Server availability and load. It monitors the operation of servers in the cluster, including the Oracle BI Scheduler instances, and is the first point of contact for requests to Oracle BI Scheduler.

The Cluster Controller supports the detection of server and Oracle BI Scheduler failures, and failover for ODBC clients of failed servers. It also determines the active Oracle BI Scheduler instance at run time.

The Cluster Controller is deployed in an active/passive configuration. The Primary Cluster Controller is the active controller, while the Secondary Cluster Controller assumes the role of the Primary Cluster Controller if the primary controller is unavailable. By default, the first Cluster Controller that is configured in your Oracle Business Intelligence installation is the Primary Cluster Controller.

Note that Presentation Services instances are not controlled by the Cluster Controller.

6.3.1.1 What if the Cluster Controller Fails?

The Cluster Controller supports detection of Oracle BI Server or Oracle BI Scheduler failures and failover for clients of failed servers.

The Cluster Controllers work on an active-passive model. All clients first attempt to connect to the Primary Cluster Controller. If the Primary Cluster Controller is unavailable, clients then connect to the Secondary Cluster Controller. The Secondary Cluster Controller then directs requests to Oracle BI Server instances based on load and availability, and to the primary Oracle BI Scheduler instance. If the Primary Cluster Controller becomes available, then all requests go to the Primary again.

The Secondary Cluster Controller monitors the session count on each Oracle BI Server, but does not dictate the active Oracle BI Scheduler unless the Primary Cluster Controller is down.

The Primary and Secondary Cluster Controllers monitor each other's life cycle. This is susceptible to a "Split-Brain" failure if the communication is down between the Cluster Controller instances, but each is up and can communicate with the other clients. In these cases, Oracle BI Servers are not affected, but the Oracle BI Scheduler might have two active instances at once. In rare cases, this might lead to double execution of jobs. When the line of communication comes back up, the Primary Cluster Controller will dictate to the cluster that only one Oracle BI Scheduler should be active. The possibility of a Split-Brain failure to occur is minimized by the fact that the Cluster components must exist on the same Local Area Network (LAN).

If both Cluster Controllers are unavailable, then Presentation Services will return an error to any new user attempting to log in. Existing sessions are not affected.

6.3.2 About Clustered Oracle BI Servers and the Master BI Server

The Oracle BI Cluster Server feature enables multiple Oracle BI Servers to act as a single server. Oracle BI Servers in the cluster share requests from multiple Oracle Business Intelligence clients.

One of the clustered Oracle BI Servers is designated as the Master BI Server. The Administration Tool connects to the master Oracle BI Server for online repository changes, using an Oracle BI ODBC DSN connection that is configured for the clustered environment. These metadata changes are then propagated to the other servers. Each slave server consumes the modified repository file when it next restarts.

In the NQClusterConfig.INI file, the parameter `MASTER_SERVER` specifies the Oracle BI Server that functions as the master server. You can also view which Oracle BI Server is the master using the Cluster Manager in the Administration Tool.

The Cluster Controller dispatches requests from clients such as Presentation Services to an active member of the Oracle BI Server cluster. The Oracle BI Server listens for client requests on a port assigned from the port range that is defined in the Scalability tab of the Capacity Management page in Fusion Middleware Control.

All Oracle BI Servers participating in a cluster can optionally share a common repository publishing directory. It holds the master copies of repositories that are edited in online mode. The clustered Oracle BI Servers examine this directory upon startup for any repository changes. The directory typically resides on a shared file system visible to all servers in the cluster. You must configure the following access to this publishing directory:

- The master server must have read and write access.
- All slave servers must have read access.

In Fusion Middleware Control, the **Shared Location** parameter on the Repository tab of the Deployment page specifies the location of the repository publishing directory.

6.3.2.1 What if a Clustered Oracle BI Server Fails?

The following are expected responses from Oracle BI Server clients during a clustered Oracle BI Server failure:

- Presentation Services

  Each Web user of Oracle Business Intelligence has requests served by one Oracle BI Server instance. If this Oracle BI Server becomes unavailable, then the user might see an error, but a browser refresh will cause a new session to be established with an available Oracle BI Server. Note that the Presentation Services component itself performs this arbitration on behalf of its users.

- Administration Tool

  The Administration Tool relays the ODBC error when the Oracle BI Server to which it is connecting becomes unavailable, and then closes the connection. The administrator must then reconnect.

- Agents

  When Oracle BI Server failure occurs, the error is relayed to the Oracle BI Scheduler, which logs the failure and retries the job. This causes a connection to be established with an available Oracle BI Server instance.

- Third-Party Clients

  Third-party clients use ODBC to connect to the Oracle BI Server. When Oracle BI Server failure occurs, the error is relayed and the session closed and reopened according to the ODBC standard.

6.3.2.2 What if the Master BI Server Fails?

The following results occur if the Master BI Server fails:

- If the Master BI Server is unavailable, then online metadata changes cannot be performed using the Administration Tool. This is an administration operation and does not impact run-time availability. Read-only access from other Oracle Business Intelligence components is unaffected, because the other components can use the slave servers.
- If the Master BI Server is permanently unavailable, then one of the other servers must be appointed as the new master, as follows:
  - Scale in the Master BI Server using the Scalability tab of the Capacity Management page in Fusion Middleware Control.
  - Restart the Oracle Business Intelligence system components.

    A new Master BI Server is assigned by the system.

6.3.3 About Clustered Oracle BI Scheduler Instances

Oracle BI Scheduler instances participate in the Oracle BI Cluster Server feature in active/passive mode. The active Oracle BI Scheduler instance processes jobs and executes agent requests. The passive Oracle BI Scheduler remains idle and does not process jobs until called on to take over in the event of an active Oracle BI Scheduler failure.

Oracle BI Scheduler listens for Cluster Controller communication and for client requests on ports assigned from the port range defined in the Scalability tab of the Capacity Management page in Fusion Middleware Control.

Note the following about Oracle BI Scheduler communication with other components:

- Oracle BI Scheduler communicates with Presentation Services for jobs such as agents that deliver alerts and reports to users. Connections for each unique user session of the agent are load balanced in round robin fashion.
- Oracle BI Scheduler is configured to communicate with the JavaHost instances in the cluster. Round robin load balancing is done for Java jobs and JavaHost extensions to agents.

6.3.3.1 What if an Oracle BI Scheduler Instance Fails?

Oracle BI Scheduler instances are monitored and managed by the Cluster Controller. If the primary Oracle BI Scheduler instance is unavailable, then the Cluster Controller activates the secondary Oracle BI Scheduler instance. If the previous primary Oracle BI Scheduler becomes available again, then the primary role does not revert.

When the primary Oracle BI Scheduler fails, any open client connections do not receive an error as the Oracle BI Scheduler protocol is stateless and seamlessly fails over.

- Agents

  Agent executions maintain state in the Oracle BI Scheduler tables. When the next instance of Oracle BI Scheduler becomes active, it reads the state of all scheduled job instances that were in progress, and executes them. A scheduled agent delivers only to those recipients that it did not deliver to before the failure of the active instance.

  Successful delivery is recorded in the Oracle BI Scheduler tables after completion of the delivery. If Oracle BI Scheduler fails over after delivery is complete, but before recording that fact, then the delivery is repeated. Because of this, a small number of repeat deliveries are possible.

  Note that agent failover is supported only for agents that are scheduled. For example, if you click **Run Agent Now** on the toolbar in Answers, and then the primary Oracle BI Scheduler fails, then that agent does not continue to run on the Secondary Scheduler.

- Java, Command Line, or Script Job

  The jobs are re-executed from the beginning with a new job instance.

> **Note:** Any job instance can be manually re-run from the Job Manager. For an agent, this delivers only to those users that did not have successful deliveries. For example, if the mail server goes down halfway through an agent's execution, the re-run of the instance delivers only to those recipients who did not receive e-mail due to the mail server crash.
>
> See Chapter 26, "Using Oracle BI Scheduler Job Manager" for information about Job Manager.

6.3.4 About the Oracle BI Cluster Server

The Oracle BI Cluster Server is comprised of the Cluster Controller, Oracle BI Server, and Oracle BI Scheduler components. All components of the BI Cluster Server feature must reside on the same Local Area Network (LAN).

You can use the Cluster Manager to monitor and manage the operations and activities of the Oracle BI Cluster Server. You can also use it to enable or quiesce Oracle BI Server clustered instances, and to manually activate the secondary Oracle BI Scheduler instance. The Cluster Manager is available in the Administration Tool when a repository is open in online mode.

6.3.4.1 About the Startup Process for the Cluster Server

The following list provides an overview of the Oracle BI Cluster Server startup process:

1. As each Oracle BI Server starts, it reads its NQSConfig.INI file and the centrally-managed parameters that are set in Fusion Middleware Control. If a server detects a syntax error while reading the NQSConfig.INI file, then it logs the error to its nqserver.log file. All syntax errors must be corrected for startup to continue.
2. Because Oracle Business Intelligence is deployed in a clustered configuration by default, each Oracle BI Server and Oracle BI Scheduler instance reads the local NQClusterConfig.INI file upon startup, even if only one Oracle BI Server or Oracle BI Scheduler is deployed. Cluster Controllers also read this file.
   - If an Oracle BI Server detects a syntax error while reading the file, then it logs the error to its nqserver.log file.
   - If a Cluster Controller detects an error while reading the file, then it logs the error to its NQCluster.log.
   - If the Oracle BI Scheduler detects an error while reading the file, then it logs the error to nqscheduler.log.
   - If a computer is hosting both an Oracle BI Server and a Cluster Controller, then messages are written to both logs.
   - All syntax errors must be corrected for startup to continue.
3. When an instance of the Oracle BI Server or Oracle BI Scheduler starts up, it waits for a connection from the Primary and Secondary Cluster Controllers.
   - The Oracle BI Server can run without the presence of a Cluster Controller instance. However, no clustered ODBC connection can be made to the Oracle BI Server if there is no running Cluster Controller service.
   - The Oracle BI Scheduler service starts, but remains in an inactive state until a Cluster Controller service instance comes online and notifies the Oracle BI Scheduler of a state change.
4. The primary and secondary Cluster Controllers begin to exchange heartbeat messages. (This step is omitted when no secondary Cluster Controller is defined.)
5. The Oracle BI Server verifies whether the repository publishing directory is available. If the repository publishing directory is not available, then the action each server takes depends on whether **Share Repository** has been selected in the Repository tab of the Deployment page in Fusion Middleware Control.
   - When **Share Repository** has been selected, if the publishing directory is not available at startup or if an error is encountered while the server is reading any of the files in the directory, then an error message is logged in the nqserver.log file and the server shuts down.
   - When **Share Repository** has not been selected, the server joins the cluster and loads the repository from its default local directory. Any changes made to the repository using the Administration Tool only affect the Oracle BI Server component that is associated with the changed repository file.
6. The primary and secondary Cluster Controllers begin to exchange heartbeat messages with each participant in the cluster.
   - The connection status is logged in the log files of the appropriate clustered instance (Oracle BI Scheduler or Oracle BI Server). Messages are also logged in the NQCluster.log file of the Cluster Controller.
   - Any participants with connection problems are not allowed to join the cluster.

   – If the server defined as the `MASTER_SERVER` for an online repository is not available, then you cannot edit the repository in online mode.

7. As each Oracle BI Server in the cluster starts, it examines the repository publishing directory for any updated repositories (when **Share Repository** has been selected in Fusion Middleware Control). This is done by comparing the date and timestamps.

   The administrator is responsible for ensuring that the time-of-day clocks are synchronized across all Oracle BI Servers and Cluster Controllers.

   – If a server detects a newer version of an existing repository, then it copies the repository to its own Repository directory.

   – A server does not detect the presence of any new repositories. A new repository must be manually propagated to all clustered servers when it is created. After that, online changes are detected at subsequent startups of each server.

8. When the Cluster Controller assigns a session to a particular Oracle BI Server, the server communicates with the back-end data source using the connection that is defined in the Connection Pool dialog for the data source. Clustered servers do not share a common connection pool. An ODBC session maintains affinity to one Oracle BI Server session for the lifetime of that session.

9. If an Oracle BI Server determines that it can satisfy all or a portion of a query from its cache file, then it does so. Clustered servers do not share a common cache. They can share cache for seeded queries, if each server instance is configured to do so.

6.3.4.2 Performance Considerations for the Cluster Server

The following list describes characteristics of the Cluster Server that might influence the performance of clustered Oracle BI Servers:

- Sessions are assigned to an Oracle BI Server when the session is established. A session is assigned to the server with the fewest sessions.
- Because each Oracle BI Server maintains its own local query results cache, data sources might receive the same query from multiple Oracle BI Servers, even though the result is cached. In addition, Oracle BI Servers that are brought online into an operational cluster might respond to queries more slowly while their local cache is being populated. If you use global caching, then queries that are explicitly seeded through a cache seeding agents are shared across nodes in the cluster.
- Because each Oracle BI Server has an independent copy of each repository and therefore its own data source connection pools, data sources might experience as many as N*M connections, where N is the number of active servers in the cluster and M is the maximum sessions that are allowed in the connection pool of a single repository. Therefore, it might be appropriate to reduce the maximum number of sessions that are configured in session pools.

6.3.5 About Clustered Presentation Services Instances

Multiple Presentation Services instances can be configured to provide scalability and high availability.

Presentation Services receives requests from the Oracle BI Presentation Services Plug-in on a port that is assigned from the port range that is defined in the Scalability tab of the Capacity Management page in Fusion Middleware Control.

Although an initial user session request can go to any Presentation Services in the cluster, each user is then bound to a specific Presentation Services instance.

Clustered Presentation Services instances can either share a common Oracle BI Presentation Catalog on a shared file system, or the catalog can be replicated across the Presentation Services instances. See Chapter 18, "Configuring and Managing the Oracle BI Presentation Catalog" for detailed information on replicating the Oracle BI Presentation Catalog.

Presentation Services communicates with other components as follows:

- Oracle BI Server: Presentation Services communicates with clustered Oracle BI Servers through the Cluster Controller to process user requests. Presentation Services uses the Oracle BI ODBC DSN that is configured for the clustered environment to identify the Primary and Secondary Cluster Controllers and the ports on which they listen. Presentation Services obtains the Oracle BI Server to which to connect from the Cluster Controller. The connection to the Oracle BI Server is established over ODBC, and subsequent requests in the same session go directly from Presentation Services to this assigned Oracle BI Server. The ODBC session between Presentation Services and the Oracle BI Server is stateful, and affinity must be maintained for the lifetime of the session.
- Oracle BI Scheduler: Presentation Services first contacts the Cluster Controller, which relays the active Oracle BI Scheduler instance. Presentation Services then establishes a session with the appropriate Oracle BI Scheduler instance. The host and Oracle instance names for the Cluster Controllers and Oracle BI Scheduler instances are specified on the Availability tab of the Capacity Management page in Fusion Middleware Control.
- JavaHost: Each Presentation Services instance is configured to communicate with multiple JavaHost instances in a cluster. The requests to the JavaHost instances are load balanced using native capability.

6.3.5.1 What if a Presentation Services Instance Fails?

Although an initial user session request can go to any Presentation Services instance, each user is then bound to a specific Presentation Services instance. Loss of that instance disconnects the session, and an error is relayed back to the browser. Any work in progress during the loss of the server that was not saved to disk is lost. The user must log in again to establish a new connection to an available Presentation Services. If user login is taking place using a Single Sign-On system such as Oracle Single Sign-On (SSO), then this re-login takes place automatically. The new Presentation Services session creates a new Oracle BI Server session.

> **Note:** When a Presentation Services instance fails, there is a small interval of time before the system recognizes that the instance has failed and before users are migrated to a new Presentation Services instance. Because of this, there might be some loss of session state.

An error is relayed to the Oracle BI Scheduler, which logs the failure and then retries the job. The retry establishes a new connection to an available Presentation Services instance.

6.3.6 About Clustered JavaHost Instances

The JavaHost component provides services that enable Oracle BI Presentation Services to support various components such as Java tasks for Oracle BI Scheduler, Oracle BI

Publisher, and graph generation. By default, Presentation Services uses its local BI JavaHost instance.

The JavaHost service receives requests from Presentation Services and Oracle BI Scheduler on a port that is assigned from the port range that is defined in the Scalability tab of the Capacity Management page in Fusion Middleware Control.

6.3.7 About Web Tier Components

Load-balanced Web servers are the entry points for Web client requests. See Section 1.8, "System Requirements and Certification" for information about supported Web servers See also "Configuring High Availability for Oracle Business Intelligence" in *Oracle Fusion Middleware High Availability Guide* for information about configuring a highly available Web tier using using Oracle HTTP Server.

An additional Oracle Business Intelligence component called the Oracle BI Presentation Services Plug-in routes session requests to Presentation Services instances using a native protocol. The connections are load balanced using native capability. Presentation Services and the Plug-in run as separate processes.

The Oracle BI Presentation Services Plug-in is one of the Oracle Business Intelligence Java components and is scaled for high availability when you add additional Managed Servers to the deployment.

6.4 Optional Configuration for Oracle Business Intelligence High Availability

Follow the steps in this section to perform optional configuration for Oracle Business Intelligence high availability.

This section contains the following topics:

- Section 6.4.1, "Setting Optional Cluster Controller Parameters"
- Section 6.4.2, "Setting Optional Presentation Services Parameters"
- Section 6.4.3, "Setting Optional Oracle BI Presentation Services Plug-in Parameters"

6.4.1 Setting Optional Cluster Controller Parameters

You can set optional parameters that are related to Cluster Controller heartbeat frequency in the NQClusterConfig.INI file.

A copy of the NQClusterConfig.INI file must reside on all computers that host a Cluster Controller, Oracle BI Server, or Oracle BI Scheduler component that participates in the cluster. You must set parameters in each copy of the file.

To set optional parameters in the NQClusterConfig.INI file:

1. Open the NQClusterConfig.INI file for editing. You can find the file at:

   ORACLE_INSTANCE/config/OracleBIApplication/coreapplication_obis*n*

2. The following cluster communication parameters are set to the default values shown. Optionally, modify the parameter values as required for the deployment.

Table 6–2 NQClusterConfig.INI Parameters for Cluster Communication

Parameter	Description	Default Value
SERVER_POLL_SECONDS	The frequency of heartbeat messages between the Cluster Controller and the Oracle BI Server and Oracle BI Scheduler nodes in the cluster.	5 seconds
CONTROLLER_POLL_SECONDS	The frequency of heartbeat messages between the Cluster Controllers.	5 seconds

3. Save and close the file.
4. Repeat these steps for every host in the deployment.
5. Restart Oracle Business Intelligence.

Example 6–1 shows example parameters in the NQClusterConfig.INI file. Note that any additional options that are not shown in this example are centrally managed and cannot be set manually.

Example 6–1 Sample Parameters for Clustering in NQClusterConfig.INI

```
#  NQClusterConfig.INI
[Cluster]
SERVER_POLL_SECONDS = 5;
CONTROLLER_POLL_SECONDS = 5;
```

6.4.2 Setting Optional Presentation Services Parameters

You can optionally configure certain parameters that control the communication between Presentation Services and the JavaHost component. To configure Presentation Services, set parameters in the instanceconfig.xml file on each computer that hosts Presentation Services.

To configure Presentation Services for clustering:

1. Open the configuration file instanceconfig.xml for editing. You can find instanceconfig.xml at:

   ORACLE_INSTANCE/config/OracleBIPresentationServicesComponent/coreapplication_obips*n*

2. Under the ServerInstance tag, the JavaHostProxy element has some optional sub-elements. Table 6–3 describes these optional sub-elements.

Table 6–3 Optional Sub-Elements for the JavaHostProxy Element

Sub-Element	Attribute	Description
LoadBalancer/Ping	keepAliveMaxFailures	Specifies the number of ping failures required before the host is declared non-functioning. The default value is 5.
LoadBalancer/Ping	keepAliveFrequencySecs	Specifies the ping frequency in seconds. The default value is 20.

3. Save and close the file.
4. Repeat these steps for every Presentation Services instance in your deployment.
5. Restart Oracle Business Intelligence.

6.4.3 Setting Optional Oracle BI Presentation Services Plug-in Parameters

You can optionally configure the Oracle BI Presentation Services Plug-in to control session redirection behavior. To do this, you must perform the steps in this section on each computer where the **analytics** Java component is installed.

To set optional parameters for the Oracle BI Presentation Services Plug-in:

1. Open the bridgeconfig.properties file for editing. You can find this file at:

   MW_HOME/user_projects/domains/*domain_name*/config/fmwconfig/biinstances/coreapplication

2. Optionally, you can include the parameter AlwaysKeepSessionAffiliation to control whether requests that belong to the same session can be redirected to another Presentation Services component if the current Presentation Services component score is too low.

   The instance score is an internal score that the load balancing algorithm associates with each Presentation Services instance in the cluster. It is based on various metrics that are collected by the load balancer.

   Set this parameter to true to disallow request redirection, or false to allow requests to be redirected. For example:

   ```
   oracle.bi.presentation.sawconnect.loadbalance.AlwaysKeepSessionAffiliation=true
   ```

3. Save and close the file.

4. Restart the **analytics** application from the Oracle WebLogic Server Administration Console. If Oracle BI Publisher is using the Oracle BI Presentation Catalog, then the **xmlpserver** application must also be restarted.

5. Repeat these steps for each computer that hosts the **analytics** Java component.

6.5 Using the Cluster Manager

The Cluster Manager in the Administration Tool was used in previous releases to monitor and manage Oracle BI Server, Oracle BI Scheduler, and Cluster Controller instances. This tool is still supported in the current release.

Although you use Fusion Middleware Control for most administrative tasks that relate to clustered components, the Cluster Manager provides a useful way to view and change the state of clustered components. For example, you can view the currently active Oracle BI Scheduler instance and change the active instance to a different Oracle BI Scheduler if necessary. You can also see which Oracle BI Server is the Master BI Server. Fusion Middleware Control shows the current status of clustered components, but does not provide a way to view or change the current state.

The Cluster Manager lets you monitor, analyze, and manage the operations of Oracle BI Server, Oracle BI Scheduler, and Cluster Controller instances in a cluster. It provides status, cache, and session information. The Cluster Manager is available only when the Administration Tool is connected to a clustered DSN.

If all Cluster Controllers or Oracle BI Servers in the cluster are currently stopped or offline, then you cannot access the Cluster Manager to start them. You must manually start one Cluster Controller (generally, the primary) and one Oracle BI Server.

The Cluster Manager window has two panes: the Explorer pane on the left side and the Information pane on the right side. The Explorer pane displays hierarchical information about the servers, schedulers, and controllers that comprise a cluster. The

Information pane shows detailed information about an item selected in the Explorer pane.

The Cluster Manager window refreshes every minute by default. You can change the interval.

To set the refresh interval for the display:

1. In the Administration Tool, open a repository in online mode.
2. Select **Manage > Clusters**.
3. Select **Refresh > Every** and choose another value from the list.
4. To refresh the display at any time, ensure that the Cluster Manager is the active window and press F5, or select **Refresh > Now**. This action retrieves the most current information for the cluster.

To activate an inactive Oracle BI Scheduler instance:

1. In the Administration Tool, open a repository in online mode.
2. Select **Manage > Clusters**.
3. In the Cluster Manager dialog, right-click an Oracle BI Scheduler instance.
4. If the Oracle BI Scheduler instance selected is inactive, then select **Activate**.

6.5.1 Viewing and Managing Cluster Information

The section describes how to view status, cache, and session information about a cluster and the meaning of the information provided.

6.5.1.1 Status Information

The Status view is automatically displayed when you first open the Cluster Manager window. You can also access the Status view by selecting **View > Status** in the Cluster Manager window.

The categories of information that are displayed in the Information pane might vary depending on the server to which the Administration Tool is connected. Table 6–4 describes categories that might appear.

Table 6–4 Status Columns

Column	Description
Last Reported Time	The time that the Cluster Controller or Oracle BI Server communicated with the Controlling Cluster Controller. If the server or controller is offline, then this field might be blank.
Name	The name of the computer that is hosting the Oracle BI Server or Cluster Controller.

Table 6–4 (Cont.) Status Columns

Column	Description
Role	The role of the object in the cluster: - **Controlling.** A Cluster Controller that is currently assigned the responsibility for control of the cluster. - **Primary.** The Primary Cluster Controller. This role is not displayed if the Primary Cluster Controller is currently the controlling Cluster Controller. - **Secondary.** The Secondary Cluster Controller. This role is not displayed if the Secondary Cluster Controller is currently the controlling Cluster Controller. - **Clustered server.** An Oracle BI Server that is a member of the cluster. This role is not displayed for the clustered server that is defined as the master server. - **Master.** The clustered server that the Administration Tool connects to for editing repositories in online mode. - **Active.** The Oracle BI Scheduler is active.
Sessions	This field is available when either Servers or an individual server is selected in the Explorer pane. It shows the number of sessions that are currently logged on to a clustered server.
Start Time	The timestamp showing when the Cluster Controller or Oracle BI Server was last started. This field is blank if the Cluster Controller or clustered server is offline.
Status	The status of the object in the cluster: - **Online.** The Cluster Controller or Oracle BI Server is online. For Cluster Controllers, this means that the controller can accept session requests and assign them to available servers within the cluster. For clustered servers, this means that the server might be assigned sessions by the Cluster Controller. - **Quiesce.** This status is applicable to clustered servers only. This means that any activity in progress on outstanding sessions is allowed to complete before the server transitions to Offline status. - **Offline.** The Cluster Controller or Oracle BI Server is offline. For Cluster Controllers, this means that the controller cannot accept session requests or assign sessions to available servers within the cluster. For clustered servers, this means that the server is not communicating with the controlling Cluster Controller and cannot accept sessions assigned by the controlling Cluster Controller. If the server subsequently becomes available, then it is allowed to participate in the cluster. To stop the Cluster Controller or clustered server after quiescing it, issue the **Stop** command. - **Forced Offline.** This status applies to clustered servers only. The Oracle BI Server has been stopped. This is identical to the offline status, except that if the Oracle BI Server comes back online, it is not assigned requests. The server remains in this state until the **Start** command is issued against this server from the Administration Tool Cluster Manager, or both Cluster Controllers are shut down and restarted. - **Online: Active.** The Oracle BI Scheduler instance is online, running, and the one to which Oracle BI Scheduler clients connect. This instance executes jobs. - **Online: Inactive.** The Oracle BI Scheduler is online but not running. This instance is ready to take over for the active instance if the active instance becomes unavailable. - **Online: Inactive Pending.** The Oracle BI Scheduler was active and is trying to go into an inactive state. This might take a few minutes (for example, if multiple jobs are running).

Table 6–4 (Cont.) Status Columns

Column	Description
Type	When Clusters is selected in the Explorer pane, this field is available. There are three types: ■ **Controller.** The object is a Cluster Controller. ■ **Server.** The object is an Oracle BI Server. ■ **Scheduler.** The object is a Scheduler Server.

6.5.1.2 Cache Information

The Cache view is available in the Cluster Manager window if caching is enabled.

The categories of information and their display sequence are controlled by the Options settings. Table 6–5 describes categories that might appear.

Table 6–5 Cache View Columns

Column	Description
Business Model	Name of the business model that is associated with the cache entry.
Column count	Number of columns in each row of this cache entry's result set.
Created	Time the result set of the cache entry was created.
Creation elapsed time	Time, in milliseconds, needed to create the result set for this cache entry.
Full size	Full size is the maximum size used, considering variable length columns, compression algorithm, and other factors. The actual size of the result set will be smaller than Full size.
Last used	Last time the result set of the cache entry satisfied a query. (After an unexpected shutdown of an Oracle BI Server, the Last used time might temporarily have a stale value, that is, older than the true value.)
Row count	Number of rows that are generated by the query.
Row size	Size of each row (in bytes) in this cache entry's result set.
SQL	Text of the SQL statement that generated the cache entry.
Use count	Number of times that this cache entry's result set has satisfied a query (since Oracle BI Server startup).
User	Name of the user who submitted the query that resulted in the cache entry.

To view cache information:

- Click an individual server in the Explorer pane, and then select **View > Cache**.

6.5.1.3 Session Information

The Session view is available for Oracle BI Servers. The information is arranged in two windows, described in Table 6–6.

- **Session window:** Appears on the top. Shows users currently logged on to the Oracle BI Server.
- **Request window:** Appears on the bottom. Shows active query requests for the user selected in the Session window.

Table 6–6 describes the information that appears in the Session window.

Table 6–6 Session Window Columns (Top Window)

Column	Description
Catalog	Name of the Oracle BI Presentation Catalog to which the session is connected.
Client Type	Type of client session. The client type of Administration is reserved for the user who is logged in with the Oracle BI Administrator user ID.
Last Active Time	Timestamp of the last activity on the session or the query.
Logon Time	Timestamp when the session logged on to the Oracle BI Server.
Repository	Logical name of the repository to which the session is connected.
Session ID	Unique internal identifier that the Oracle BI Server assigns each session when the session is initiated.
User	Name of the user connected.

Table 6–7 describes the information that appears in the Request window.

Table 6–7 Request Window Columns (Bottom Window)

Column	Description
Last Active Time	Timestamp of the last activity on the session or the query.
Request ID	Unique internal identifier that the Oracle BI Server assigns each query when the query is initiated.
Session ID	Unique internal identifier that the Oracle BI Server assigns each session when the session is initiated.
Start Time	Time of the initial query request.
Status	These are the possible values. Due to the speed at which some processes complete, not all values for any given request or session might appear. ■ **Idle.** There is presently no activity on the request or session. ■ **Fetching.** The request is being retrieved. ■ **Fetched.** The request has been retrieved. ■ **Preparing.** The request is being prepared for processing. ■ **Prepared.** The request has been prepared for processing and is ready for execution. ■ **Executing.** The request is currently running. To terminate a request, select it and click **Kill Request**. The user receives an informational message that indicates that the Oracle BI Administrator canceled the request. ■ **Executed.** The request has finished running. ■ **Succeeded.** The request ran to completion successfully. ■ **Canceled.** The request has been canceled. ■ **Failed.** An error was encountered during the processing or running of the request.

To manage clustered servers:

1. In the Explorer pane, expand the **Server** icon to display the servers in the cluster.
2. In the Information pane, select a server.
3. Select **Action**, and then select one of the available options.

When the operation finishes, the status of the clustered server is refreshed automatically.

To view session information:

1. Select a server in the Explorer pane, and then select **View > Sessions**.

   Session information for the server is displayed in the Information pane. It shows all users logged into the server and all current query requests for each user.

To disconnect a session:

1. In the Session view, right-click the session in the Session window (top window) and click **Disconnect**.

   When you disconnect a session, the ODBC session is terminated. Client users who were connected over this session receives errors if they attempt to run queries. Users must log out, then log back in again to start a new session.

To terminate a query request:

1. In the Session view, right-click the request in the Request window (bottom window) and click **Kill Request**.

   When you terminate a query request, the user who is initiating the query receives an error.

6.5.1.4 Server Information

Selecting Server info from the View menu provides information about the cluster server, such as server version number.

6.6 Troubleshooting an Oracle Business Intelligence Clustered Environment

Use Fusion Middleware Control and the Administration Console to check the status of system processes. See Section 5.6.1, "Using Fusion Middleware Control to View System Component Availability" and Section 5.6.2, "Using the Administration Console to View Managed Server Availability" for more information.

After enabling clustering, load balancing, and failover capabilities, you can troubleshoot issues that might occur in the deployment using the following:

- Messages and errors that are reported in Fusion Middleware Control
- Log files for Oracle Business Intelligence components, also available through Fusion Middleware Control

Review the log files for every Oracle Business Intelligence system component in the cluster. Log files record any client-side failures that might occur due to misconfiguration. While some failover events are not logged, the Cluster Controller log file records crashes of any Oracle BI Scheduler or Oracle BI Server component. You can also review the Event Viewer log on Windows and the syslog on Linux or UNIX.

See Chapter 8, "Diagnosing and Resolving Issues in Oracle Business Intelligence" for more information about log files.

6.6.1 Avoiding Errors with Network Appliance Devices when the Oracle BI Server is Running on Linux or UNIX

The following information applies to deployments with Oracle BI Server components on Linux or UNIX platforms that access Oracle Business Intelligence shared files and directories on a NAS device from Network Appliance. For environments with Oracle BI Server components on Linux or UNIX that use the NTFS security style, the recommended Network Appliance Data ONTAP storage operating system version is 6.3.1 or higher.

Linux or UNIX computers saving to an NTFS qtree in Data ONTAP versions 6.0.3 through 6.3 might see permission errors when trying to save designs. Use the following Data ONTAP setting to silently ignore attempts to set UNIX permissions on NTFS qtrees after the design file is saved:

```
options cifs.ntfs_ignore_unix_security_ops on
```

7 Managing Performance Tuning and Query Caching

This chapter provides information about ways to improve Oracle Business Intelligence query performance, including a performance tuning overview and information about monitoring system metrics. It also describes the how to manage and use the query cache, a feature that enables the Oracle BI Server to save the results of a query in cache files and then reuse those results later when a similar query is requested. Using cache, the cost of database processing only needs to be paid once for a query, not every time the query is run.

This chapter contains the following topics:

7.1 Monitoring Service Levels

Understanding service levels typically involves monitoring process state and viewing system metrics.

Oracle Business Intelligence automatically and continuously measures run-time performance in real time. The performance metrics are automatically enabled; you do not need to set options or perform any extra configuration to collect them.

System metrics are available in Fusion Middleware Control for system components within a given Oracle Business Intelligence installation. If you encounter a problem, such as an application that is running slowly or is hanging, then you can view more detailed performance information to learn more information about the problem.

You can use WSLT commands to periodically save metric information to a file so that you have a record of past metric values. See "DMS Custom WLST Commands" in

Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for more information.

You can also view metrics for Java components using the Oracle WebLogic Server Administration Console.

This section contains the following topics:

- Section 7.1.1, "Using Fusion Middleware Control to View Common Performance Metrics"
- Section 7.1.2, "Using Fusion Middleware Control to View All Oracle Business Intelligence Metrics"
- Section 7.1.3, "Using the Administration Console to View Metrics for Java Components"

7.1.1 Using Fusion Middleware Control to View Common Performance Metrics

You can access the most commonly viewed performance metrics from the Metrics tab of the Capacity Management page.

To use Fusion Middleware Control to view common performance metrics:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Metrics tab of the Capacity Management page.
3. On the Metrics tab, you can view metrics that are related to responsiveness, load, and reliability. Click the **Help** button on the page to access the page-level help for the following metrics:
   - Request Processing Time (ms)
   - SOA Request Processing Time (ms)
   - Average Query Time (seconds)
   - Active Sessions
   - Requests (per minute)
   - SOA Requests (per minute)
   - Presentation Server Requests (per second)
   - Server Queries (per second)
   - Failed Queries
   - Errors Reported (in the last hour)

   The metrics that are displayed on this tab enable you to determine the current responsiveness, load, and reliability for Oracle Business Intelligence components across the entire cluster.

7.1.2 Using Fusion Middleware Control to View All Oracle Business Intelligence Metrics

You can view and graph all the available Oracle Business Intelligence metrics from the Performance Summary page in Fusion Middleware Control. The data is logged transiently (that is, logging starts when you go to the page and select a particular metric for display).

To use Fusion Middleware Control to view all performance metrics for Oracle Business Intelligence:

1. In the tree navigator, expand the Business Intelligence folder and right-click the coreapplication node.
2. Select **Monitoring**, then select **Performance**. The Performance Summary page appears, displaying a selection of metrics for this Oracle Business Intelligence installation.

   Note: Alternatively, to view the Performance Summary page, you can go to the Metrics page of the Capacity Management tab and click **View the full set of system metrics**.

3. To customize the metrics that are displayed on the Performance Summary page, click Show Metric Palette. Then, expand the appropriate metric category and select or deselect individual metrics. The metrics that you select are displayed on the Performance Summary page.

   For information about a particular metric, right-click the metric and select **Help**.

7.1.3 Using the Administration Console to View Metrics for Java Components

Use the Administration Console to view metrics for Java components. You can view metrics on the Monitoring tab for the selected Managed Server, or you can use the Metric Browser.

To view metrics for Oracle Business Intelligence Managed Servers:

1. Log in to the Administration Console.
2. Expand the **Environment** node in the Domain Structure window.
3. Click **Servers**. The Summary of Servers page is displayed.
4. Click the Managed Server name (for example, **oracle_bi1**).
5. Click the Monitoring tab.

   Click **Help** for more information about the metrics displayed on this tab.

To access the Administration Console Metric Browser:

1. Log in to the Administration Console.
2. Click **Monitoring Dashboard** under Charts and Graphs.
3. Click the Metric Browser tab.

   Click **Help** for more information about using the Metric Browser.

7.2 About Query Performance Tuning

This section describes some important considerations for improving query performance with the Oracle BI Server.

The following list summarizes methods that you can use to improve query performance:

- **Tuning and indexing underlying databases:** For Oracle BI Server database queries to return in a timely manner, the underlying databases *must* be configured, tuned, and indexed correctly. Note that different database products have different tuning considerations.

If there are queries that return slowly from the underlying databases, then you can capture the SQL statements for the queries in the query log and provide them to your DBA for analysis. See Section 8.3, "Managing the Query Log" for more information about configuring query logging on the system.

- **Aggregate tables:** It is extremely important to use aggregate tables to improve query performance. Aggregate tables contain precalculated summarizations of data. It is much faster to retrieve an answer from an aggregate table than to recompute the answer from thousands of rows of detail.

  The Oracle BI Server uses aggregate tables automatically, if they have been properly specified in the repository. See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for examples of setting up aggregate navigation.

- **Query caching:** The Oracle BI Server can store query results for reuse by subsequent queries. Query caching can dramatically improve the apparent performance of the system for users, particularly for commonly-used dashboards, but it does not improve performance for most ad-hoc analysis.

  See Section 7.4, "About the Oracle BI Server Query Cache" for more information about query caching concepts and setup.

- **Setting parameters in Fusion Middleware Control:** You can set various performance configuration parameters using Fusion Middleware Control to improve system performance. See Section 7.3, "Setting Performance Parameters in Fusion Middleware Control" for more information.

- **Setting parameters in NQSConfig.INI:** The NQSConfig.INI file contains additional configuration and tuning parameters for the Oracle BI Server, including parameters to configure disk space for temporary storage, set virtual table page sizes, and a number of other advanced configuration settings. See Appendix A, "NQSConfig.INI File Configuration Settings" for more information.

You can also improve the overall performance of the system by increasing throughput by scaling out system components. See Chapter 5, "Scaling Your Deployment" for more information.

7.3 Setting Performance Parameters in Fusion Middleware Control

This section describes performance options that you can set in Fusion Middleware Control.

This section contains the following topics:

- Section 7.3.1, "Using Fusion Middleware Control to Disallow RPD Updates"
- Section 7.3.2, "Using Fusion Middleware Control to Set the User Session Log-Off Period"
- Section 7.3.3, "Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables"
- Section 7.3.4, "Using Fusion Middleware Control to Set the Maximum Number of Rows Processed to Render a Table"

7.3.1 Using Fusion Middleware Control to Disallow RPD Updates

You can use Fusion Middleware Control to allow or disallow updates to the default repository file. Setting this parameter affects whether you can update the repository when the Administration Tool connects in both online and offline mode. It also affects whether you can perform other repository update operations using other utilities, such as biserverxmlcli. Note that aggregate persistence feature is not available when repository updates are disallowed.

Disallowing repository updates can improve Oracle BI Server performance, because in this mode, the Oracle BI Server does not need to handle lock control.

If you choose to disallow repository updates, then when the Administration Tool opens a repository in either online or offline mode, a message informs the user that the repository is read-only.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to disallow repository updates:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Performance tab of the Capacity Management page.
3. Click **Lock and Edit Configuration** to allow changes to be made.
4. Select **Disallow Online RPD Updates** to disallow updates to the repository file.

   Click the **Help** button on the page to access the page-level help.
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

7.3.2 Using Fusion Middleware Control to Set the User Session Log-Off Period

You can override the time to elapse, in minutes, before a user is automatically logged off. Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to set the client session log-off period:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Performance tab of the Capacity Management page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the **User Session Expiry** option using the description in the Help topic for the page.

   Click the **Help** button on the page to access the page-level help for the box.

5. Click **Apply**, then click **Activate Changes** to execute your changes and release the lock to enable another system administrator to make changes.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to change settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

7.3.3 Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables

Advanced configuration settings are described in Section 19.3, "Configuring for Displaying and Processing Data in Views."

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to set configuration options for views:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Performance tab of the Capacity Management page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
   - **Maximum Number of Rows to Download to Excel** option
   - **Maximum Number of Rows Per Page to Include in Email** option
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

7.3.4 Using Fusion Middleware Control to Set the Maximum Number of Rows Processed to Render a Table

You can override the maximum number of rows that can be fetched and processed from the Oracle BI Server for rendering a table. Reducing the number of rows in a table can significantly improve performance by reducing the system resources that can be consumed by a given user session.

Advanced configuration settings are described in Section 19.3, "Configuring for Displaying and Processing Data in Views."

Note the following when setting this value:

- This specification applies to tables, not to pivot tables.
- The default value is 65000. The minimum value is 50. If the user exceeds the maximum value, then the server returns an error message when the table view is rendered. The maximum value is at least 16 bits, which varies by platform. The

system will likely consume all its memory before approaching a number larger than this value.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to set the maximum number of rows that are processed to render a table:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Performance tab of the Capacity Management page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the **Maximum Number of Rows Processed to Render A Table View** option using the description in the Help topic for the page. Enter an integer value greater than 50.

   Click the **Help** button on the page to access the page-level help for the box.
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

7.4 About the Oracle BI Server Query Cache

You can configure the Oracle BI Server to maintain a local, disk-based cache of query result sets (query cache). The query cache allows the Oracle BI Server to satisfy many subsequent query requests without having to access back-end data sources (such as Oracle or DB2). This reduction in communication costs can dramatically decrease query response time.

As updates occur on the back-end databases, the query cache entries can become stale. Therefore, you must periodically remove entries from the query cache using one of the following methods:

- **Manually.** In the Oracle BI Administration Tool, in the **Manage** menu, select **Cache** to open the Cache Manager. The Cache Manager provides the most flexibility in choosing which cache entries to purge and when to purge them, but it requires manual intervention. See Section 7.7.4, "Using the Cache Manager" for more information.
- **Automatically.** In the Administration Tool, you can disable cache for the system, set caching attributes for a specific physical table, and use Oracle Business Intelligence event tables to purge cache automatically. See Section 7.6, "Monitoring and Managing the Cache" for additional information.
- **Programmatically.** The Oracle BI Server provides ODBC-extension functions for purging cache entries programmatically. These functions give you the choice and the timing flexibility of the Cache Manager with the automation of event tables. You can write your own scripts to call these functions at times that fit your needs. See Section 7.6.2, "Purging and Maintaining Cache Using ODBC Procedures" for more information.

The parameters that control query caching are located in Fusion Middleware Control and in the NQSConfig.INI file, described in Appendix A, "NQSConfig.INI File Configuration Settings." See also Section 7.7.3, "Using Agents to Seed the Oracle BI Server Cache" for additional information.

This section contains the following topics:

- Section 7.4.1, "Query Cache Architecture"
- Section 7.4.2, "Advantages of Caching"
- Section 7.4.3, "Costs of Caching"
- Section 7.4.4, "Cache Sharing Across Users"
- Section 7.4.5, "About the Refresh Interval for XML Data Sources"
- Section 7.4.6, "About the Global Cache"

7.4.1 Query Cache Architecture

The query cache consists of cache storage space, cache metadata, and cache detection in query compilation.

The process of the Oracle BI Server accessing the cache metadata is very fast. If the metadata shows a cache hit, then the bulk of the query processing is eliminated, and the results are immediately returned to the user. The process of adding the new results to the cache is independent of the results being returned to the user; the only effect on the running query is the resources that are consumed in the process of writing the cached results.

Query cache entries are portable across different operating systems, such as Windows or UNIX, and across 32-bit and 64-bit architectures. Incompatible cache entries are automatically removed. For example, you do not have to manually remove cache files when switching between 32-bit and 64-bit systems.

Note that query cache entries are *not* portable across different releases of Oracle Business Intelligence, such as between Version 10.1.3.2 and 11*g* Release 1 (11.1.1).

Caching occurs by default at the subrequest level, which results in multiple cache entries for some SQL statements. Caching subrequests improves performance and the cache hit ratio, especially for queries that combine real-time and historical data. To disable subrequest caching, set the NQSConfig.INI file parameter `DISABLE_SUBREQUEST_CACHING` to `YES`. See Appendix A, "NQSConfig.INI File Configuration Settings" for more information.

7.4.2 Advantages of Caching

The fastest way to process a query is to skip the bulk of the processing and use a precomputed answer. With query caching, the Oracle BI Server stores the precomputed results of queries in a local cache. If another query can use those results, then all database processing for that query is eliminated. This can result in dramatic improvements in the average query response time.

In addition to improving performance, being able to answer a query from a local cache conserves network resources and processing time on the database server. Network resources are conserved because the intermediate results do not have to come over the network to the Oracle BI Server. Not running the query on the database frees the database server to do other work. If the database uses a charge back system, then it could save money in the budget as well.

Another benefit of using the cache to answer a query is savings in processing time on the Oracle BI Server, especially if the query results are retrieved from multiple databases. Depending on the query, there might be considerable join and sort processing in the server. If the query is already calculated, then this processing is avoided, freeing server resources for other tasks.

To summarize, query caching has the following advantages:

- Dramatic improvement of query performance
- Less network traffic
- Reduction in database processing
- Reduction in Oracle BI Server processing overhead

7.4.3 Costs of Caching

Query caching has many obvious benefits, but also certain costs:

- Disk space for the cache
- Administrative costs of managing the cache
- Potential for cached results being stale
- CPU and disk I/O on server computer

With cache management, the benefits typically far outweigh the costs.

The following sections discuss the costs of caching.

7.4.3.1 Disk Space

The query cache requires dedicated disk space. How much space depends on the query volume, the size of the query result sets, and how much disk space that you choose to allocate to the cache. For performance purposes, a disk should be used exclusively for caching, and it should be a high performance, high reliability type of disk system.

7.4.3.2 Administrative Tasks

There are a few administrative tasks that are associated with caching. You must set the cache persistence time for each physical table appropriately, knowing how often data in that table is updated. When the frequency of the update varies, you must keep track of when changes occur and purge the cache manually when necessary. You can also create a cache event polling table and modify applications to update the polling table when changes to the databases occur, making the system event-driven.

The Oracle BI Server also provides ODBC-extension functions for purging cache entries programmatically. You can write your own scripts to call these functions at the appropriate times.

7.4.3.3 Keeping the Cache Up To Date

If the cache entries are not purged when the data in the underlying databases changes, then queries can potentially return results that are out of date. You must evaluate whether this is acceptable. It might be acceptable to allow the cache to contain some stale data. You must decide what level of stale data is acceptable and then configure (and follow) a set of rules to reflect those levels.

For example, suppose an application analyzes corporate data from a large conglomerate, and you are performing yearly summaries of the different divisions in the company. New data does not materially affect the queries because the new data affects only next year's summaries. In this case, the trade-offs for deciding whether to purge the cache might favor leaving the entries in the cache.

Suppose, however, that the databases are updated three times a day and you are performing queries on the current day's activities. In this case, you must purge the cache much more often, or perhaps consider not using the cache at all.

Another scenario is that you rebuild the data mart from scratch at periodic intervals (for example, once per week). In this example, you can purge the entire cache as part of the process of rebuilding the data mart, ensuring that you never have stale data in the cache.

Whatever your situation, you must evaluate what is acceptable for noncurrent information returned to the users.

7.4.3.4 CPU Usage and Disk I/O

Although in most cases it is very minor, query caching does require a small amount of CPU time and adds to the disk I/O. In most cases, the CPU usage and disk I/O is insignificant. The disk I/O might be noticeable only when queries return large data sets.

7.4.4 Cache Sharing Across Users

If shared logon has been enabled for a particular connection pool, then the cache can be shared across users and does not need to be seeded for each user. If shared logon has not been enabled and a user-specific database login is used, then each user generates their own cache entries.

See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about enabling shared logon for connection pools.

7.4.5 About the Refresh Interval for XML Data Sources

Typically, XML data sources are updated frequently and in real time. Setting a refresh interval for XML data sources is analogous to setting cache persistence for database tables. The refresh interval is a time interval after which the XML data sources are to be queried again directly, rather than using results in cache. This refresh interval is specified on the XML tab of the Connection Pool dialog.

The default interval setting is **Infinite**, meaning that the XML data source is not automatically refreshed.

The refresh interval setting determines the time interval after which the Oracle BI Server XML Gateway connection is refreshed, as follows:

- For URLs that begin with `http://` or `https://`, the gateway is refreshed when it detects that the interval has expired.
- For URLs that reside on a local or network drive, the gateway is refreshed when the interval has expired and the system detects that the URLs have been modified.

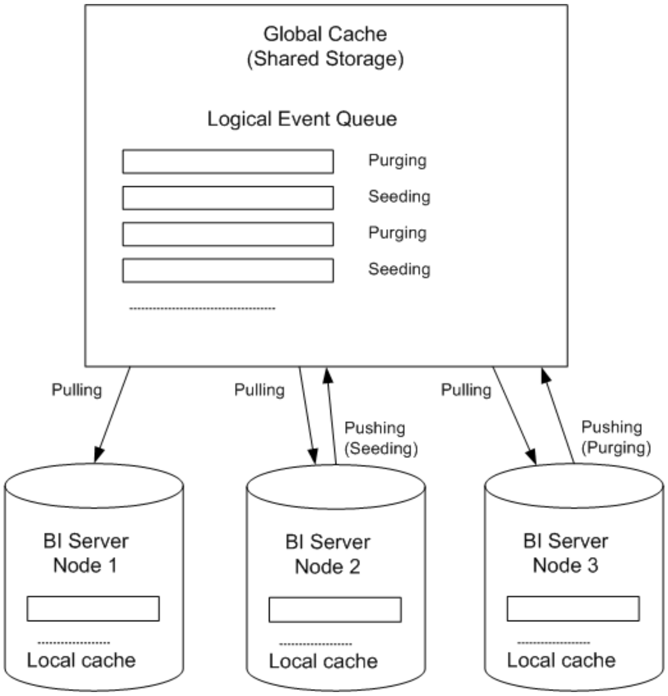
For more information about XML data sources, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

7.4.6 About the Global Cache

In a clustered environment, Oracle BI Servers can be configured to access a shared cache called the global cache. This global cache resides on a shared file system storage device and stores purging events, seeding events (often generated by agents), and result sets that are associated with seeding events. The seeding and purging events are sorted by time and stored on the shared storage as a logical event queue. Individual Oracle BI Server nodes push to and pull from the logical event queue. Each Oracle BI Server still maintains its own local query cache for regular queries.

Figure 7–1 depicts global caching in a clustered environment. It shows three Oracle BI Server nodes sharing a global cache. The global cache stores seeding or purging events held in a logical event queue. The arrows from Node 2 and Node 3 to the shared cache show Oracle BI Server Node 2 pushing a seeding event to the queue and Oracle BI Server Node 3 pushing a purging event to the queue. The arrows from the shared storage to each Oracle BI Server node show each node pulling from the common location. This occurs on a periodic basis and enables participating Oracle BI Server nodes to obtain updates to the logical event queue made by other Oracle BI Servers.

Figure 7–1 Global Caching



The Oracle BI Server node processes a seeding or purging event locally first in its caching system. It then pushes the event to the global cache on the shared storage. During the push event, the active Oracle BI Server node locks the logical event queue on the shared storage and then pushes in the seeding or purging event. If there is a conflict between seeding and purging (for example, one node wants to seed a query and another node wants to purge the same query), then the event that comes in last wins.

The logical event queue in the global cache on the shared storage is composed of seeding and purging events from individual Oracle BI Server nodes. The queue is sorted according to the timestamp of the events. Hence, clocks on all Oracle BI Server nodes participating in cluster must be synchronized.

Each Oracle BI Server node polls the global cache on a periodic basis for new cache entries. This polling frequency is configurable. A snapshot of the queued logical events on the shared storage are pulled back to the node and a local logical event queue is constructed and then processed.

> **Note:** The process of populating or purging seeded caches across all Oracle BI Server nodes that participate in the cluster does not occur in real time, and the elapse of the process is affected by multiple factors, such as the predefined polling interval, network bandwidth, and CPU loads.

Because the query cache result set tends to get large, network bandwidth might pose a constraint. Therefore, the following must be chosen carefully:

- The set of caches that qualify for seeded cache
- The time interval for BI nodes to pick up seeded caches from shared storage (to avoid network congestion)

The primary global cache parameters are configured in Fusion Middleware Control. Additional, optional parameters are configured in the NQSConfig.INI file for each Oracle BI Server node that participates in the cluster. For more information about configuring these parameters, see Section 7.5.4, "Using Fusion Middleware Control to Set Global Cache Parameters" and Section 7.5.5, "Manually Editing Additional Global Cache Parameters."

A seeding or purging procedure is submitted to a specific Oracle BI Server node. If that Oracle BI Server is a node in a BI cluster and the global cache parameters have been defined in Oracle BI Server configuration files, then the seeding or purging events are propagated across all Oracle BI Server nodes that participate in the same clustered environment.

7.5 Configuring Query Caching

You configure cache storage and other parameters in Fusion Middleware Control and in the NQSConfig.INI file, for both the query cache and the global cache. You also must decide on a strategy for flushing outdated cache entries; see Section 7.6, "Monitoring and Managing the Cache" for more information.

This section contains the following topics:

- Section 7.5.1, "Using Fusion Middleware Control to Enable and Disable Query Caching"
- Section 7.5.2, "Using Fusion Middleware Control to Set Query Cache Parameters"
- Section 7.5.3, "Manually Editing Additional Query Cache Parameters"
- Section 7.5.4, "Using Fusion Middleware Control to Set Global Cache Parameters"
- Section 7.5.5, "Manually Editing Additional Global Cache Parameters"

7.5.1 Using Fusion Middleware Control to Enable and Disable Query Caching

You can use Fusion Middleware Control to enable or disable query caching. The query cache is enabled by default.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to enable or disable query caching:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Performance tab of the Capacity Management page.
3. Click **Lock and Edit Configuration** to allow changes to be made.
4. To enable query caching, select **Cache enabled**. To disable query caching, deselect **Cache enabled**.

   Click the **Help** button on the page to access the page-level help.
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

7.5.2 Using Fusion Middleware Control to Set Query Cache Parameters

You can use Fusion Middleware Control to set the maximum number of cache entries in the query cache and the maximum size for a single cache entry.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to set query cache parameters:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Performance tab of the Capacity Management page.
3. Click **Lock and Edit Configuration** to allow changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:

   - **Maximum cache entry size**
   - **Maximum cache entries**
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

7.5.3 Manually Editing Additional Query Cache Parameters

You can set additional query cache parameters in the NQSConfig.INI file, including the following:

- The `DATA_STORAGE_PATHS` parameter specifies one or more directories for query cache storage, and the maximum size for each storage directory. These directories are used to store the cached query results and are accessed when a cache hit occurs. See Section 7.7.1, "About Cache Hits" for more information about when cache is hit.

  The cache storage directories should reside on high performance storage devices, ideally devoted solely to cache storage. When the cache storage directories begin to fill up, the entries that are least recently used (LRU) are discarded to make space for new entries.

- The `MAX_ROWS_PER_CACHE_ENTRY` parameter controls the maximum number of rows for any cache entry. Limiting the number of rows is a useful way to avoid using up the cache space with runaway queries that return large numbers of rows. If the number of rows a query returns is greater than the value specified in the `MAX_ROWS_PER_CACHE_ENTRY` parameter, then the query is not cached.

- Typically, if a query gets a cache hit from a previously executed query, then the new query is not added to the cache. The `POPULATE_AGGREGATE_ROLLUP_HITS` parameter overrides this default when the cache hit occurs by rolling up an aggregate from a previously executed query.

See Appendix A, "NQSConfig.INI File Configuration Settings" for more information about the additional query cache parameters.

7.5.4 Using Fusion Middleware Control to Set Global Cache Parameters

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to set global cache parameters:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Performance tab of the Capacity Management page.
3. Click **Lock and Edit Configuration** to allow changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
   - **Global cache path**
   - **Global cache size**
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

7.5.5 Manually Editing Additional Global Cache Parameters

You can set additional global cache parameters in the NQSConfig.INI file, including the following:

- The `MAX_GLOBAL_CACHE_ENTRIES` parameter controls the maximum number of entries that are allowed in the global cache store.
- The `CACHE_POLL_SECONDS` parameter specifies the interval in seconds at which the Oracle BI Server pulls from the logical event queue to synchronize with other server nodes in the cluster.
- The `CLUSTER_AWARE_CACHE_LOGGING` parameter controls whether logging is turned on for the global cache. Change this setting to `YES` only for debugging purposes.

  Log entries appear in nqquery.log. You can find this file at:

  ```
  ORACLE_INSTANCE\diagnostics\logs\OracleBIServerComponent\coreapplication_obisn
  ```

See Appendix A, "NQSConfig.INI File Configuration Settings" for more information about the additional global cache parameters.

7.6 Monitoring and Managing the Cache

To manage the changes in the underlying databases and to monitor cache entries, you must develop a cache management strategy. You need a process to invalidate cache entries when the data in the underlying tables that compose the cache entry have changed, and a process to monitor, identify, and remove any undesirable cache entries.

This section contains the following topics:

- Section 7.6.1, "Choosing a Cache Management Strategy"
- Section 7.6.2, "Purging and Maintaining Cache Using ODBC Procedures"
- Section 7.6.3, "How Repository Changes Affect the Query Cache"

7.6.1 Choosing a Cache Management Strategy

The choice of a cache management strategy depends on the volatility of the data in the underlying databases and the predictability of the changes that cause this volatility. It also depends on the number and types of queries that comprise your cache and the usage those queries receive. This section provides an overview of the various approaches to cache management.

7.6.1.1 Disable Caching for the System

You can disable caching for the entire system to stop all new cache entries and stop any new queries from using the existing cache. Disabling caching lets you enable it at a later time without losing any entries that are already stored in the cache.

Temporarily disabling caching is a useful strategy in situations where you might suspect having stale cache entries, but want to verify if they are actually stale before purging those entries or the entire cache. If you find that the data stored in the cache is still relevant, or after you have safely purged problem entries, then you can safely enable the cache. If necessary, purge the entire cache or the cache that is associated with a particular business model before enabling the cache again.

See Section 7.5.1, "Using Fusion Middleware Control to Enable and Disable Query Caching" for more information.

7.6.1.2 Caching and Cache Persistence Timing for Specified Physical Tables

You can set a cacheable attribute for each physical table, enabling you to specify whether queries for that table are added to the cache to answer future queries. If you enable caching for a table, then any query involving the table is added to the cache. All tables are cacheable by default, but some tables might not be good candidates to include in the cache unless you use the Cache Persistence Time settings. For example, suppose that you have a table that stores stock ticker data that is updated every minute. You could use the Cache Persistence Time settings to purge the entries for that table every 59 seconds.

You can also use the **Cache persistence time** field to specify how long the entries for this table should be kept in the query cache. This is useful for data sources that are updated frequently.

To set the caching attributes for a specific physical table:

1. In the Administration Tool, in the Physical layer, double-click the physical table.
2. In the Physical Table properties dialog, in the General tab, make one of the following selections:
   - To enable caching, select **Cacheable**.
   - To prevent a table from being cached, deselect **Cacheable**.
3. To set a cache expiration time, specify a **Cache persistence time** and specify a unit of measure (days, hours, minutes, or seconds). If you do not want cache entries to automatically expire, select **Cache never expires**.
4. Click **OK**.

7.6.1.3 Configure Oracle BI Server Event Polling Tables

Oracle BI Server event polling tables store information about updates in the underlying databases. An application (such as one that loads data into a data mart) could be configured to add rows to an event polling table each time a database table is updated. The Oracle BI Server polls this table at set intervals and invalidates any cache entries corresponding to the updated tables. Event polling tables can be the sole method of cache management, or they can be used with other cache management schemes. Event tables offer less flexibility about choice of cache entries and the timing of purges. See Section 7.8.1, "Setting Up Event Polling Tables on the Physical Databases" for more information about event polling tables.

7.6.2 Purging and Maintaining Cache Using ODBC Procedures

The Oracle BI Server provides ODBC-extension functions for purging cache entries.

Some of these functions are particularly useful for embedding in an Extract, Transform, and Load (ETL) task. For example, after a nightly ETL is performed, all Oracle BI Server cache entries can be purged. If only the fact table was modified, then only cache related to that table can be purged. In some cases, you might need to purge the cache entries associated with a specific database.

Only administrators have the right to purge cache. Therefore, scripts that call these ODBC-extension functions must run under credentials with administrator privileges.

The following ODBC functions affect cache entries that are associated with the repository specified by the ODBC connection:

- **SAPurgeCacheByQuery.** Purges cache entries that exactly match a specified query. For example, using the following query, you would have one or more query cache entries that retrieve the names of all employees earning more than $100,000:

  ```
  SELECT lastname, firstname FROM employee WHERE salary > 100000;
  ```

  The following call purges the cache entries that are associated with this query:

  ```
  Call SAPurgeCacheByQuery('SELECT lastname, firstname FROM employee WHERE salary
  > 100000' );
  ```

- **SAPurgeCacheByTable.** Purges all cache entries that are associated with a specified physical table name (fully qualified) for the repository to which the client has connected.

  This function takes up to four parameters that represent the four components (database, catalog, schema, and table name proper) of a fully qualified physical table name. For example, you might have a table with the fully qualified name of `DBName.CatName.SchName.TabName`. To purge the cache entries that are associated with this table in the physical layer of the Oracle Business Intelligence repository, run the following call in a script:

  ```
  Call SAPurgeCacheByTable( 'DBName', 'CatName', 'SchName', 'TabName' );
  ```

  > **Note:** Wildcards are not supported by the Oracle BI Server for this function. In addition, DBName and TabName cannot be null. If either one is null, then an error message is displayed.

- **SAPurgeAllCache.** Purges all cache entries. The following is an example of this call:

  ```
  Call SAPurgeAllCache();
  ```

- **SAPurgeCacheByDatabase.** Purges all cache entries associated with a specific physical database name. A record is returned as a result of calling any of the ODBC procedures to purge the cache. This function takes one parameter that represents the physical database name, and the parameter cannot be null. The following shows the syntax of this call:

  ```
  Call SAPurgeCacheByDatabase( 'DBName' );
  ```

7.6.2.1 About ODBC Procedure Syntax

If there is a single quote within the string argument of a procedure, then you must use another single quote to escape it. For example:

```
Call SAPurgeCacheByQuery('SELECT TOPN("- Currency"."Markdown %", 10) saw_0,
"XX Line"."Order No" saw_1, "- Bill-To Site"."Customer Name" saw_2, "-
Currency"."Net USD" saw_3, "- Currency"."Markdown USD" saw_4, "-
Currency"."Markdown %" saw_5 FROM "Apps 11i - XX Lines" WHERE
("XX Line"."Open Flag" = ''Y'') AND ("Operating Unit"."Group Name" = ''Group'')
AND ("- Currency"."Net USD" >= 10000) ORDER BY saw_0');
```

The line in bold highlights the extra single quotes used as escape characters for the items **`''Y''`** and **`''Group''`**.

7.6.2.2 About Sharing the Presentation Services Query Cache

When users access Answers to run queries, Presentation Services caches the results of the queries. Presentation Services uses the request key and the logical SQL string to determine if subsequent queries can use cached results. If the cache can be shared, then subsequent queries are not stored.

- **SAGetSharedRequestKey:** An ODBC procedure that takes a logical SQL statement from Presentation Services and returns a request key value.

  The following shows the syntax of this procedure:

  ```
  SAGetSharedRequestKey('sql-string-literal')
  ```

The value of the request key is affected by the following factors:

- Whether the **Virtual Private Database** option has been selected in the repository physical database object
- Whether any session variables have been marked as **Security Sensitive** in the repository

Presentation Services takes security sensitive variable values into consideration when computing the request key for logical requests against database objects marked as Virtual Private Databases.

See Section 7.9, "Managing the Oracle BI Presentation Services Cache Settings" for more information about the Presentation Services query cache.

7.6.2.3 About Result Records

A result record is returned after you issue a purge cache command. The result record contains two columns. The first column is a result code and the second column is a short message that describes the result of the purge operation. Table 7–1 shows examples of result records.

Table 7–1 Query Result Codes

Result Code	Result Message
1	SAPurgeCacheByDatabase returns successfully.
59115	Operation not performed because caching is not enabled.
59116	The database specified does not exist.
59117	The table specified does not exist.

7.6.2.4 Storing and Purging Cache for SAP/BW Data Sources

In Microsoft Analysis Services, member caption name is the same as member unique name. However, in SAP/BW data sources, member caption name is different from member unique name. Therefore, the Oracle BI Server maintains a cache subsystem for SAP/BW member unique names. This subsystem is turned off by default. For configuration information, see the topic about the MDX Member Name Cache Section in Appendix A, "NQSConfig.INI File Configuration Settings."

When a query is received for member unique name, the subsystem checks the cache to determine whether cache exists for this query. If cache exists, then the record for the cached unique name is returned. If there is no cache that matches the query, then the subsystem sends a probing query to SAP/BW.

The probing query is logged when the log level is equal or greater than 2. The status of the subsystem, such as if the subsystem is enabled and events such as start and shutdown events, are also written to the server log.

Caution: With each increased logging level, performance is impacted. Use caution when increasing the log level for users.

Be aware of the following cache purge issues:

- The size of multidimensional cache entries can grow very large. Therefore, a limit on the size of each member set has been established in the `MDX_MEMBER_CACHE` section of the NQSConfig.INI file.
- The format of persisted cache might not be consistent after an upgrade. Therefore, you should purge all cache before a software upgrade.
- The cache is populated the first time that the query runs. You should arrange to populate the cache during off-peak hours, to minimize performance impact.

Note: In the Administration Tool, you can purge cache for an individual cube table by right-clicking the cube table, and then selecting **Purge Member Cache**. This must be performed in online mode by a user with administrator privileges.

The following purge procedures are specific to SAP/BW data sources:

- **SAPurgeALLMCNCache.** Purges all SAP/BW cache entries.

  The following shows the syntax of this procedure:

  ```
  SAPurgeALLIMCNCache ()
  ```

- **SAPurgeMCNCacheByCube.** Purges all cache entries that are associated with the specified physical cube. The database name and cube name are the external names of the repository objects. The following shows the syntax of this procedure:

  ```
  SAPurgeMCNCacheByCube( 'DBName', 'CubeName')
  ```

The following messages is returned.

Table 7–2 SAP Purge Cache Return Codes and Messages

Return Code	Return Message
1	SAPurgeALLMCNCache returns successfully.
1	SAPurgeMCNCacheByCube returns successfully.
59116	The database specified does not exist. **Note:** If the database and physical cube are both wrong, then this result code is returned.
85025	The physical cube specified does not exist.

Only users with administrative privileges can run ODBC purge procedures.

7.6.3 How Repository Changes Affect the Query Cache

When you modify Oracle Business Intelligence repositories, the changes can have implications for entries that are stored in the cache. For example, if you change the definition of a physical object or a dynamic repository variable, cache entries that reference that object or variable might no longer be valid. These changes might result in the need to purge the cache. There are three scenarios to be aware of: when the changes occur in online mode, when they occur in offline mode, and when you are switching between repositories.

7.6.3.1 Online Mode

When you modify an Oracle Business Intelligence repository in online mode, any changes that you make that affect cache entries automatically result in a purge of all cache entries that reference the changed objects. The purge occurs when you check in the changes. For example, if you delete a physical table from a repository, then all cache entries that reference that table are purged upon check in. Any changes made to a business model in the Business Model and Mapping layer purge all cache entries for that business model.

7.6.3.2 Offline Mode

When you modify an Oracle Business Intelligence repository in offline mode, you might make changes that affect queries that are stored in the cache and render those cached results obsolete. Because the repository is not loaded by the server during offline mode edits, the server has no way of determining if the changes made affect any cached entries. The server therefore does not automatically purge the cache after offline changes. If you do not purge the cache, then there might be invalid entries when the repository is next loaded. Unless you are sure that there are no entries in the cache that are affected by your offline changes, then you should purge the cache for any business model that you have modified.

7.6.3.3 Switching Between Repositories

If you intend to remove a repository from the configuration of the Oracle BI Server, then ensure that you purge the cache of all cache entries that reference the repository. Failure to do so results in a corrupted cache. See Section 7.7.4.2, "Purging Cache in the Administration Tool" for more information.

7.6.3.4 Changes to Dynamic Repository Variables

The values of dynamic repository variables are refreshed by data that is returned from queries. When you define a dynamic repository variable, you create an initialization block or use a preexisting one that contains a SQL query. You also configure a schedule for the Oracle BI Server to follow to execute the query and periodically refresh the value of the variable.

When the value of a dynamic repository variable changes, all cache entries that are associated with a business model that reference the value of that variable are purged automatically. The cache entries are purged when the repository variable refresh rate is reached, if its value has changed.

Note that if a business model is not associated with a changed dynamic repository variable, then no cache purging action occurs. For example, suppose an initialization block has been defined with a repository variable and a refresh rate of 5 minutes. But, no logical column has been defined that references the variable. When the value of the dynamic repository variable changes, cache is not purged because no logical column exists within a business model that uses the variable.

7.7 Strategies for Using the Cache

One of the main advantages of query caching is to improve apparent query performance. It might be valuable to seed the cache during off hours by running queries and caching their results. A good seeding strategy requires that you know when cache hits occur.

If you want to seed the cache for all users, then you might seed the cache with the following query:

```
SELECT User, SRs
```

After seeding the cache using `SELECT User, SRs`, the following queries will be cache hits:

```
SELECT User, SRs WHERE user = valueof(nq_SESSION.USER) (and the user was USER1)
SELECT User, SRs WHERE user = valueof(nq_SESSION.USER) (and the user was USER2)
SELECT User, SRs WHERE user = valueof(nq_SESSION.USER) (and the user was USER3)
```

This section contains the following topics:

- Section 7.7.1, "About Cache Hits"
- Section 7.7.2, "Running a Suite of Queries to Populate the Cache"
- Section 7.7.3, "Using Agents to Seed the Oracle BI Server Cache"
- Section 7.7.4, "Using the Cache Manager"

7.7.1 About Cache Hits

When caching is enabled, each query is evaluated to determine whether it qualifies for a cache hit. A cache hit means that the server was able to use cache to answer the query and did not go to the database at all. The Oracle BI Server can use the query cache to answer queries at the same or higher level of aggregation.

Many factors determine whether cache is hit. Table 7–3 describes these factors.

Table 7–3 Factors That Determine Whether Cache Is Hit

Factor or Rule	Description
A subset of columns in the `SELECT` list must match	All of the columns in the `SELECT` list of a new query have to exist in the cached query to qualify for a cache hit, or they must be able to be calculated from the columns in the query. This rule describes the minimum requirement to hit the cache, but meeting this rule does not guarantee a cache hit. The other rules listed in this table also apply.

Columns in the `SELECT` list can be composed of expressions on the columns of the cached queries	The Oracle BI Server can calculate expressions on cached results to answer the new query, but all the columns must be in the cached result. For example, the query:
	`SELECT product, month, averageprice FROM sales WHERE year = 2000`
	will hit cache on the query:
	`SELECT product, month, dollars, unitsales FROM sales WHERE year = 2000`
	because `averageprice` can be computed from `dollars` and `unitsales` (`averageprice = dollars/unitsales`).

`WHERE` clause must be semantically the same or a logical subset	For the query to qualify as a cache hit, the `WHERE` clause constraints must be either equivalent to the cached results, or a subset of the cached results.
	A `WHERE` clause that is a logical subset of a cached query qualifies for a cache hit if the subset meets one of the following criterion:
	■ A subset of `IN` list values. Queries requesting fewer elements of an `IN` list cached query qualify for a cache hit. For example, the following query:
	`SELECT employeename, region`
	`FROM employee, geography`
	`WHERE region in ('EAST', 'WEST')`
	qualifies as a hit on the following cached query:
	`SELECT employeename, region`
	`FROM employee, geography`
	`WHERE region in ('NORTH', 'SOUTH', 'EAST', 'WEST')`
	■ It contains fewer (but identical) `OR` constraints than the cached result.
	■ It contains a logical subset of a literal comparison. For example, the following predicate:
	`WHERE revenue < 1000`
	qualifies as a cache hit on a comparable query with the predicate:
	`WHERE revenue < 5000`
	■ There is no `WHERE` clause. If a query with no `WHERE` clause is cached, then queries that satisfy all other cache hit rules qualify as cache hits regardless of their `WHERE` clause.

Dimension-only queries must be an exact match	If a query is dimension only, meaning that no fact or measure is included in the query, then only an exact match of the projection columns of the cached query hits the cache. This behavior prevents false positives when there are multiple logical sources for a dimension table.
Queries with special functions must be an exact match	Other queries that contain special functions such as time series functions (`AGO`, `TODATE`, and `PERIODROLLING`), external aggregation functions, and filter metrics must also be an exact match with the projection columns in the cached query. In these cases, the filter must also be an exact match.
Set of logical tables must match	To qualify as a cache hit, all incoming queries must have the same set of logical tables as the cache entry. This rule avoids false cache hits. For example, `SELECT * FROM product` does not match `SELECT * FROM product, sales`.

Table 7–3 (Cont.) Factors That Determine Whether Cache Is Hit

Factor or Rule	Description
Session variable values must match, including security session variables	If the logical SQL or physical SQL statement refers to any session variable, then the session variable values must match. Otherwise, the cache is not hit. In addition, the value of session variables that are security sensitive must match the security session variable values that are defined in the repository, even though the logical SQL statement itself does not reference session variables. See "Ensuring Correct Cache Results When Using Row-Level Database Security" for more information.
Equivalent join conditions	The resultant joined logical table of a new query request has to be the same as (or a subset of) the cached results to qualify for a cache hit.
`DISTINCT` attribute must be the same	If a cached query eliminates duplicate records with `DISTINCT` processing (for example, `SELECT DISTINCT`...), then requests for the cached columns must also include the `DISTINCT` processing; a request for the same column without the `DISTINCT` processing is a cache miss.

Queries must contain compatible aggregation levels	Queries that request an aggregated level of information can use cached results at a lower level of aggregation. For example, the following query requests the quantity sold at the supplier and region and city level:
	`SELECT supplier, region, city, qtysold`
	`FROM suppliercity`
	The following query requests the quantity sold at the city level:
	`SELECT city, qtysold`
	`FROM suppliercity`
	The second query results in a cache hit on the first query.

Limited additional aggregation	For example, if a query with the column `qtysold` is cached, then a request for `RANK(qtysold)` results in a cache miss. Additionally, a query that requests `qtysold` at the country level can get a cache hit from a query that requests `qtysold` at the country, region level.
`ORDER BY` clause must be made up of columns in the select list	Queries that order by columns that are not contained in the select list result in cache misses.
Avoiding cache misses using advanced hit detection	You can avoid some cache misses by setting the parameter `USE_ADVANCED_HIT_DETECTION` to `YES` in the NQSConfig.INI file. Advanced hit detection enables an expanded search of the cache for hits. See "USE_ADVANCED_HIT_DETECTION" for more information.

7.7.1.1 Ensuring Correct Cache Results When Using Row-Level Database Security

When using a row-level database security strategy, such as a Virtual Private Database (VPD), the returned data results are contingent on the authorization credentials of the user. Because of this, the Oracle BI Server must know whether a data source is using row-level database security and which variables are relevant to security.

To ensure that cache hits only occur on cache entries that include and match all security-sensitive variables, you must correctly configure the database object and session variable objects in the Administration Tool, as follows:

- **Database object.** In the Physical layer, in the General tab of the Database dialog, select **Virtual Private Database** to specify that the data source is using row-level database security.

  If you are using row-level database security with shared caching, then you *must* select this option to prevent the sharing of cache entries whose security-sensitive variables do not match.

- **Session Variable object.** For variables that you are using for authentication, in the Session Variable dialog, select **Security Sensitive** to identify them as sensitive to security when using a row-level database security strategy. This option ensures that cache entries are marked with the security-sensitive variables, enabling security-sensitive variable matching on all incoming queries.

Refer to the following resources for more information:

- "Setting Up Row-Level Security in the Database" in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*
- "Managing Session Variables" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for general information about database and session variable objects

7.7.2 Running a Suite of Queries to Populate the Cache

To maximize potential cache hits, one strategy is to run a suite of queries just for the purpose of populating the cache. The following are some recommendations for the types of queries to use when creating a suite of queries with which to seed the cache.

- **Common prebuilt queries.** Queries that are commonly run, particularly ones that are expensive to process, are excellent cache seeding queries. Queries whose results are embedded in dashboards are good examples of common queries.
- **SELECT lists with no expressions.** Eliminating expressions on `SELECT` list columns expands the possibility for cache hits. A cached column with an expression can only answer a new query with the same expression; a cached column with no expressions can answer a request for that column with any expression. For example, a cached request such as:

  ```
  SELECT QUANTITY, REVENUE...
  ```

  can answer a new query such as:

  ```
  SELECT QUANTITY/REVENUE...
  ```

  but not the reverse.
- **No WHERE clause.** If there is no `WHERE` clause in a cached result, then it can be used to answer queries that satisfy the cache hit rules for the select list with any `WHERE` clause that includes columns in the projection list.

In general, the best queries to seed cache with are queries that heavily consume database processing resources and that are likely to be reissued. Be careful not to seed the cache with simple queries that return many rows. These queries (for example, `SELECT * FROM PRODUCTS`, where `PRODUCTS` maps directly to a single database table) require very little database processing. Their expense is network and disk overhead, which are factors that caching does not alleviate.

When the Oracle BI Server refreshes repository variables, it examines business models to determine if they reference those repository variables. If they do, the Oracle BI Server then purges all cache for those business models. See Section 7.6.3.4, "Changes to Dynamic Repository Variables" for more information.

7.7.3 Using Agents to Seed the Oracle BI Server Cache

You can configure agents to seed the Oracle BI Server cache. Seeding the cache can improve response times for users when they run analyses or view analyses that are embedded on their dashboards. You can accomplish this by scheduling agents to execute requests that refresh this data.

To configure an agent to seed the Oracle BI Server cache:

1. Log in to Oracle Business Intelligence and select **New**, then select **Agent**.
2. On the General tab, select **Recipient** for the **Run As** option. Personalized cache seeding uses the data visibility of each recipient to customize agent delivery content for each recipient.
3. On the Schedule tab, specify when you want the cache to be seeded.
4. Optionally, select Condition and create or select a conditional request. For example, you might have a business model that determines when the ETL process is complete. You could use a report based on this business model to be the conditional trigger for the cache seed to begin.
5. On the Delivery Content tab, select an individual request or an entire dashboard page for which you want to seed the cache. Selecting a dashboard page can save time.
6. On the Recipients tab, select individual users or groups to be the recipients.
7. On the Destinations tab, clear all user destinations and select **Oracle BI Server Cache**.
8. Save the agent by selecting the **Save** button in the upper-right corner.

The only difference between cache seeding agents and other agents is that they clear the previous cache automatically and do not appear on the dashboard as Alerts.

Note that cache seeding agents only purge exact match queries, so stale data might still exist. Your caching strategy should always include cache purging, because agent queries do not address ad-hoc queries or drills.

7.7.4 Using the Cache Manager

The Cache Manager lets you view information about the entire query cache and information about individual entries in the query cache that are associated with the open repository. You can also use it to select specific cache entries and perform various operations on those entries, such as viewing and saving the cached SQL statement, or purging them.

To open the Cache Manager:

1. In the Administration Tool toolbar, select **Manage > Cache**.

Select the Cache tab on the left explorer pane to view the cache entries for the current repository, business models, and users. The associated cache entries are reflected in the right pane, with the total number of entries shown in the view-only field at the top.

The cache entry information and its display sequence is controlled by the Options settings (select **Edit**, then select **Options** from the Cache Manager, or select **Tools > Options > Cache Manager** from the Administration Tool menu). Information can include the options that are described in Table 7–4.

Table 7–4 Cache Options

Option	Description
User	The ID of the user who submitted the query that resulted in the cache entry.
Created	The time the cache entry's result set was created.
Last used	The last time the cache entry's result set satisfied a query. (After an unexpected shutdown of the Oracle BI Server, the last used time might temporarily have a stale value—a value that is older than the true value.)
Creation elapsed time	The time, in seconds, that is needed to create the result set for this cache entry. **Note:** The value that is stored in the cache object descriptor on disk is in units of milliseconds. The value is converted to seconds for display purposes.
Row count	The number of rows generated by the query.
Row size	The size of each row (in bytes) in this cache entry's result set.
Full size	Full size is the maximum size used, considering variable length columns, compression algorithm, and other factors. The actual size of the result set will be smaller than Full size.
Column count	The number of columns in each row of this cache entry's result set.
Logical Request	The logical request that is associated with this cache entry. If subrequests are being cached, then this column shows the text of the subrequest.
Use count	The number of times that this cache entry's result set has satisfied a query (since Oracle BI Server startup).
Business model	The name of the business model that is associated with the cache entry.
Repository	The name of the Oracle Business Intelligence repository that is associated with this cache entry.
SQL	The SQL statement that is associated with this cache entry. If subrequests are being cached, then there might be multiple cache entries that are associated with a single SQL statement.
Query Server	The Oracle BI Server that serviced the query.
Fact Table Source	The fact table that is associated with the logical request for this cache entry.

Expand the repository tree to display all the business models with cache entries, and expand the business models to display all users with cache entries. The right pane displays only the cache entries associated with the selected item in the hierarchical tree.

7.7.4.1 Displaying Global Cache Information in the Cache Manager

Select **Action**, then select **Show Info** to display global cache information. Table 7–5 describes the information that appears in the Global Cache Information window.

Table 7–5 Global Cache Information

Column	Description
Amount of space still available for cache storage use	The amount of space, in megabytes, still available for cache storage.
Amount of space used on disks containing cache related files	The total amount of space, in megabytes, used on the disk that contains cache-related files (not just space used for the cache-related files).
Maximum allowable number of entries in cache	The maximum number of entries that can be in the cache, from the `MAX_CACHE_ENTRIES` parameter in the NQSConfig.INI file.
Maximum allowable number of rows per cache entry result set	The maximum number of rows that are allowed for each cache entry's result set, from the `MAX_ROWS_PER_CACHE_ENTRY` parameter in the NQSConfig.INI file.
Number of entries currently in cache	The current number of entries in the global cache. These entries might relate to multiple repositories.
Number of queries not satisfied from cache since startup of Oracle BI Server	Cache misses, since the last time the Oracle BI Server was started.
Number of queries satisfied from cache since startup of Oracle BI Server	Cache hits, since the last time the Oracle BI Server was started.

With the Cache Manager as the active window, press F5, or select **Action > Refresh** to refresh the display. This retrieves the current cache entries for the repository that you have open and the current global cache information. If the DSN is clustered, then information about all repositories in the cluster is displayed.

7.7.4.2 Purging Cache in the Administration Tool

Purging cache is the process of deleting entries from the query cache. You can purge cache entries in the following ways:

- Manually, using the Administration Tool Cache Manager facility (in online mode).
- Automatically, by setting the **Cache Persistence Time** field in the Physical Table dialog for a particular table.
- Automatically, by setting up an Oracle BI Server event polling table.
- Automatically, as the cache storage space fills up.

> **Note:** You can also purge the cache programmatically using ODBC-extension functions. See Section 7.6.2, "Purging and Maintaining Cache Using ODBC Procedures" for more information.
>
> In addition, cache can be purged when the value of dynamic repository variables changes. See Section 7.6.3.4, "Changes to Dynamic Repository Variables" for more information.

To manually purge cache entries in the Cache Manager:

1. Use the Administration Tool to open a repository in online mode.
2. Select **Manage > Cache** to open the Cache Manager dialog.
3. Select Cache or Physical mode by selecting the appropriate tab in the left pane.

4. Browse the explorer tree to display the associated cache entries in the right pane.
5. Select the cache entries to purge, and then select **Edit > Purge** to remove them. Or, right-click the selected entries and then select **Purge**.
   - In Cache mode, select the entries to purge from those displayed in the right pane.
   - In Physical mode, select the database, catalog, schema or tables to purge from the explorer tree in the left pane.

   In Cache mode, you can purge:

   - One or more selected cache entries that are associated with the open repository.
   - One or more selected cache entries that are associated with a specified business model.
   - One or more selected cache entries that are associated with a specified user within a business model.

   In Physical mode, you can purge:

   - All cache entries for all tables that are associated with one or more selected databases.
   - All cache entries for all tables that are associated with one or more selected catalogs.
   - All cache entries for all tables that are associated with one or more selected schemas.
   - All cache entries that are associated with one or more selected tables.

Purging deletes the selected cache entries and associated metadata. Select **Action > Refresh** or press F5 to refresh your cache display.

7.8 Cache Event Processing with an Event Polling Table

You can use an Oracle BI Server event polling table (event table) as a way to notify the Oracle BI Server that one or more physical tables have been updated. Each row that is added to an event table describes a single update event, such as an update occurring to the Product table in the Production database. The Oracle BI Server cache system reads rows from, or polls, the event table, extracts the physical table information from the rows, and purges stale cache entries that reference those physical tables.

The event table is a physical table that resides on a database accessible to the Oracle BI Server. Regardless of whether it resides in its own database, or in a database with other tables, it requires a fixed schema, that is described in Table 7–6. It is normally exposed only in the Physical layer of the Administration Tool, where it is identified in the Physical Table dialog as an Oracle BI Server event table.

Using event tables is one of the most accurate ways of invalidating stale cache entries, and it is probably the most reliable method. It does, however, require the event table to be populated each time that a database table is updated. Also, because there is a polling interval in which the cache is not completely up to date, there is always the potential for stale data in the cache. See Section 7.8.3, "Populating the Oracle BI Server Event Polling Table" for more information.

A typical method of updating the event table is to include SQL `INSERT` statements in the extraction and load scripts or programs that populate the databases. The `INSERT` statements add one row to the event table each time that a physical table is modified.

After this process is in place and the event table is configured in the Oracle Business Intelligence repository, cache invalidation occurs automatically. As long as the scripts that update the event table are accurately recording changes to the tables, stale cache entries are purged automatically at the specified polling intervals.

This section contains the following topics:

- Section 7.8.1, "Setting Up Event Polling Tables on the Physical Databases"
- Section 7.8.2, "Making the Event Polling Table Active"
- Section 7.8.3, "Populating the Oracle BI Server Event Polling Table"
- Section 7.8.4, "Troubleshooting Problems with Event Polling Tables"

7.8.1 Setting Up Event Polling Tables on the Physical Databases

This section describes how to configure the Oracle BI Server event polling tables on physical databases.

7.8.1.1 Polling Table Structure

You can configure a physical event polling table on each physical database to monitor changes in the database. You can also configure the event table in its own database. The event table should be updated every time a table in the database changes.

If the event polling table is on an Oracle Database, you should configure the event table in its own database object in the Physical layer of the Administration Tool. Then, ensure that the feature PERF_PREFER_IN_LISTS is not selected in the Features tab of the Database dialog for the event polling table. Following these guidelines avoids errors related to exceeding the maximum number of allowed expressions in a list.

Event tables must have the structure that is shown in Table 7–6. Some columns can contain null values, depending on where the event table resides. The names for the columns must match the column names that are shown in Table 7–6.

See Section 7.8.1.2, "Sample Event Polling Table CREATE TABLE Statements" for samples. Alternatively, you can use the sample event polling table in the database that you installed for use with Oracle Business Intelligence.

Table 7–6 Event Polling Table Column Names

Event Table Column	Data Type	Description
CatalogName	CHAR or VARCHAR	The name of the catalog where the physical table that was updated resides. Populate the CatalogName column only if the event table does not reside in the same database as the physical tables that were updated. Otherwise, set it to the null value.

Table 7–6 (Cont.) Event Polling Table Column Names

Event Table Column	Data Type	Description
DatabaseName	CHAR or VARCHAR	The name of the database where the physical table that was updated resides. This is the name of the database as it is defined in the Physical layer of the Administration Tool. For example, if the physical database name is 11308Production, and the database name that represents it in the Administration Tool is SQL_Production, then the polled rows in the event table must contain SQL_Production as the database name. Populate the DatabaseName column only if the event table does not reside in the same database as the physical tables that were updated. Otherwise, set it to the null value.
Other	CHAR or VARCHAR	Reserved for future enhancements. This column must be set to a null value.
SchemaName	CHAR or VARCHAR	The name of the schema where the physical table that was updated resides. Populate the SchemaName column only if the event table does not reside in the same database as the physical tables being updated. Otherwise, set it to a null value.
TableName	CHAR or VARCHAR	The name of the physical table that was updated. The name must match the name that is defined for the table in the Physical layer of the Administration Tool. Values cannot be null.
UpdateTime	DATETIME	The time when the update to the event table occurs. This must be a key (unique) value that increases for each row that is added to the event table. To ensure a unique and increasing value, specify the current timestamp as a default value for the column. For example, specify `DEFAULT CURRENT_TIMESTAMP` for Oracle Database. Values cannot be null. **Note:** Because this column must be a unique value that increases for each row that is added to the event table, you might need to set a very high precision if you require many inserts per second. Because of this, you might want to adjust the database feature `FRACTIONAL_SECOND_PRECISION` to allow fractional seconds to be used in the filters on the UpdateTime column. The Oracle BI Server truncates the timestamps to the number of digits that are defined by `FRACTIONAL_SECOND_PRECISION`. For example, for Oracle Database or Teradata, you might want to change `FRACTIONAL_SECOND PRECISION` from 0 to 6.
UpdateType	INTEGER	Specify a value of 1 in the update script to indicate a standard update. (Other values are reserved for future use.) Values cannot be null.

The Oracle BI Server must have read and write permission on the event polling table. The server reads the event table at specified intervals to look for changed data. Applications add rows to the event table when database tables are modified (for example, during a load operation). When there are rows in the event table, there is changed data in the underlying databases. The server then invalidates any cache entries that correspond to the changed physical tables and periodically deletes obsolete rows from the event table. The next time it checks the event table, the process repeats.

Note: In a clustered Oracle Business Intelligence deployment, a single event polling table is shared by every Oracle BI Server node in the cluster. However, a single event polling table cannot be shared by multiple Oracle BI Server clusters.

To enable the Oracle BI Server to have write access to the event polling table but not to any other tables in a database, perform the following tasks:

- Create a separate physical database in the Physical layer of the Administration Tool with a privileged connection pool.
- Assign a user to the connection pool that has delete privileges.
- Populate the privileged database with the event table.

The Oracle BI Server has write access to the event polling table, but not to any tables that are used to answer user queries.

7.8.1.2 Sample Event Polling Table CREATE TABLE Statements

This section provides sample `CREATE TABLE` statements for SQL Server and Oracle Database. These `CREATE TABLE` statements create the structure that is required for an Oracle BI Server event polling table. In these statements, the table that is created is named UET. It resides in the same database as the physical tables that are being updated.

Note: The column lengths must be large enough to represent the object names in the repository.

The following is a sample `CREATE TABLE` statement for SQL Server:

```
// SQL Server Syntax
create table UET (
  UpdateType Integer not null,
  UpdateTime datetime not null DEFAULT CURRENT_TIMESTAMP,
  DBName     char(40) null,
  CatalogName varchar(40) null,
  SchemaName  varchar(40) null,
  TableName   varchar(40) not null,
  Other       varchar(80) null DEFAULT NULL
)
```

The following is a sample `CREATE TABLE` statement for Oracle Database:

```
// Oracle Database syntax
create table UET (
  UpdateType Integer not null,
  UpdateTime date DEFAULT SYSDATE not null,
  DBName     char(40) null,
  CatalogName varchar(40) null,
  SchemaName  varchar(40) null,
  TableName   varchar(40) not null,
  Other       varchar(80) DEFAULT NULL
);
```

You might need to modify these `CREATE TABLE` statements slightly for different versions of SQL Server and Oracle Database, or for other databases. Additionally, if you want to specify any explicit storage clauses, then you must add the appropriate clauses to the statements. Similarly, if you need international characters, then you must modify the DDL to handle international characters.

7.8.2 Making the Event Polling Table Active

After the table is created on the physical database, you can make it active in the Oracle BI Server. To do this, you first import the physical table, and then you mark the table object as an event polling table.

To import the physical table:

1. In the Administration Tool, open the repository and import metadata from the physical database. To do this, select **File**, then select **Import Metadata**.
2. Follow the wizard steps. Be sure to select the **Tables** option in the Select Metadata Types screen to import the table metadata.

   See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for detailed information about the Import Metadata wizard.
3. If you have multiple event polling tables, then repeat steps 1 and 2 for each event table. Be sure the data source that is specified for the event table has read and write access to the event table. The repository both reads the table and deletes rows from it, so it needs write permission. Event tables do not need to be exposed in the Business Model and Mapping layer.

To mark the table object as an event polling table:

1. Select the **Tools > Utilities** menu item.
2. Select the option **Oracle BI Event Tables** from the list of options.
3. Click **Execute**.
4. Select the table that you want to register as an Event Table and click the **>>** button.
5. Specify the polling frequency in minutes, and click **OK**.

   The default value is 60 minutes. You should not set the polling frequency to less than 10 minutes. If you want a very short polling interval, then consider marking some or all of the tables non cacheable.

When a table has been registered as an Oracle BI Server event table, the table properties change. Registration as an event table removes the option to make the table cacheable, as there is no reason to cache results from an event polling table.

7.8.3 Populating the Oracle BI Server Event Polling Table

The Oracle BI Server does not populate the event polling table. The event table is populated by inserting rows into it each time that a table is updated. This process is normally configured by the database administrator, who typically modifies the load process to insert a row into the polling table each time a table is modified. This can be done from the load script, using database triggers (in databases that support triggers), from an application, or manually. If the process of populating the event table is not done correctly, then the Oracle BI Server cache purging is affected, because the server assumes the information in the polling table is correct and up to date.

7.8.4 Troubleshooting Problems with Event Polling Tables

If you experience problems with cache polling, then you can search the Oracle BI Server activity logs for any entries regarding the server's interaction with the event table.

- The nqserver.log file logs activity automatically about the Oracle BI Server. Log entries are self-explanatory and can be viewed in Fusion Middleware Control or in a text editor.
- When the Oracle BI Server polls the event table, it logs queries in the nqquery.log file using the administrator account (set upon installation) unless the logging level for the administrator account is set to 0. You should set the logging level to 2 for the administrator account to provide the most useful level of information.

You can find the nqserver.log and the nqquery.log in the following location:

```
ORACLE_INSTANCE/diagnostics/logs/OracleBIServerComponent/coreapplication_obisn
```

7.9 Managing the Oracle BI Presentation Services Cache Settings

When users run analyses, Presentation Services can cache the results of those analyses. Presentation Services determines if subsequent analyses can use cached results. If the cache can be shared, then subsequent analyses are not stored.

The files for the Presentation Services cache have names such as nQS_*xxxx_x_xxxxxx*.TMP. The files are created by the ODBC driver but generally do correspond to ODBC requests that the Presentation Services cache keeps open. The files are stored in the following directory:

ORACLE_INSTANCE\tmp\OracleBIPresentationServices\coreapplication_obips*n*\obis_temp

The files for the cache are removed whenever Presentation Services shuts down cleanly. If Presentation Services shuts down unexpectedly, then various cache files might be left on disk. You can delete the files when Presentation Services is not running.

The Presentation Services cache is not the same cache that is accessed by the Oracle BI Server. You can change the defaults for the Presentation Services cache by modifying the instanceconfig.xml file to include the cache entries.

The following procedure provides information on configuration changes with which you can manage the Presentation Services cache.

See Section 7.6.2.2, "About Sharing the Presentation Services Query Cache" for information on an ODBC procedure to use for sharing the cache.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings for managing the cache:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the elements that are described in Table 7–7.

> **Note:** Avoid specifying values of less than 3 minutes for the elements that affect minutes. At such a low amount of time, refreshes can occur frequently, which can negatively affect performance and cause flickering on the screen.

3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <Cache>
    <Query>
      <MaxEntries>100</MaxEntries>
      <MaxExpireMinutes>60</MaxExpireMinutes>
      <MinExpireMinutes>10</MinExpireMinutes>
      <MinUserExpireMinutes>10</MinUserExpireMinutes>
    </Query>
  </Cache>
<ServerInstance>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

Table 7–7 Elements for Configuring the Cache for Presentation Services

Element	Description	Default Value
MaxEntries	Specifies the maximum number of open record sets that Presentation Services keeps open at any one time. The minimum value is 3. For systems under significant loads, you can increase this value to 700 or 1000.	500
MaxExpireMinutes	Specifies the maximum amount of time, in minutes, that an entry in the cache can exist before it is removed. Depending on the number of analyses being run, an entry might be removed before the time limit expires.	60 (one hour)
MinExpireMinutes	Specifies the minimum amount of time, in minutes, that an entry in the cache can exist before it is removed. The setting for CacheMinUserExpireMinutes can force an entry for a particular user to exist for a longer time than that specified by the CacheMaxExpireMinutes element.	10
MinUserExpireMinutes	Specifies the minimum amount of time, in minutes, that an entry in the cache can exist after it has been viewed by a user. For example, if CacheMaxExpireMinutes is set to 60 minutes and a user views the entry during the 59th minute, the entry exists for that user for an additional 10 minutes. The user can continue paging through the data without requiring a new analysis to be run.	10

7.10 Improving Oracle BI Web Client Performance

This section describes recommendations to improve performance for the Oracle Business Intelligence Web client.

This section contains the following topics:

- Section 7.10.1, "Setting Up Static File Caching"
- Section 7.10.2, "Bypassing Static Files"

7.10.1 Setting Up Static File Caching

You can improve the performance of the Oracle BI Web client by caching small, frequently used static files such as .javascript, .gif, and .css files. By enabling caching and content expiration on the Web server, Web browsers can determine how often to reload the static files from the server.

This section contains the following topics:

- Section 7.10.1.1, "Setting Up Static File Caching for Microsoft IIS Server"
- Section 7.10.1.2, "Setting Up Static File Caching for Oracle HTTP Server/Apache HTTP Server"

7.10.1.1 Setting Up Static File Caching for Microsoft IIS Server

Follow these steps to configure static file caching and content expiration if you are using Microsoft IIS Server with Oracle Business Intelligence.

To set up static file caching for Microsoft IIS Server:

1. On the Web server computer, from the Start menu, select **Settings**, then **Control Panel**, then **Administrative Tools**.
2. Run Internet Service Manager.
3. In Internet Service Manager, right-click the **Default Web Site**. You can specify content expiration at the individual Web site folder level, the virtual directory level, or for a file.
4. In Default Web Site Properties, click the HTTP Headers tab.
5. Select **Enable Content Expiration**.
6. Select **Expire After**, and specify the value appropriate for your deployment (for example, seven).
7. Restart IIS.

7.10.1.2 Setting Up Static File Caching for Oracle HTTP Server/Apache HTTP Server

Follow these steps to configure static file caching and content expiration if you are using Oracle HTTP Server or Apache HTTP Server with Oracle Business Intelligence.

To set up static file caching for Oracle HTTP Server or Apache HTTP Server:

1. On the Web server computer, open the file httpd.conf for editing.

   This file is located in the Web server installation directory (for example, *ORACLE_HOME*/Apache/Apache/conf).

2. Verify that the following directive is included and not commented out:

   ```
   LoadModule expires_module modules/mod_expires.so
   ```

Note: For Apache versions prior to 1.3.15 on Windows, the directive is LoadModule expires_module modules/ApacheModuleExpires.dll.

3. Add the following lines to the file below the directive specified in step 2:

```
ExpiresActive On
<IfModule mod_expires.c>
ExpiresByType image/gif "access plus 7 days"
ExpiresByType image/jpeg "access plus 7 days"
ExpiresByType application/x-javascript "access plus 7 days"
ExpiresByType text/css "access plus 7 days"
</IfModule>
```

Note: In this example, content is set to expire in seven days. Set a value that is appropriate for your deployment.

4. Save the file.
5. Restart the HTTP Server.

7.10.2 Bypassing Static Files

Performance can be improved by configuring the HTTP Server to serve the static files. By default, the static file requests for Oracle Business Intelligence are served by the WebLogic Server.

Because the Oracle Business Intelligence static files reside outside the HTTP Server's document root, you must configure the HTTP Server to access the files from a file system that is not the document root for the HTTP Server. Consult your vendor documentation for more information on configuring the HTTP Server to access files not residing in the document root.

On Apache HTTP Servers, the Alias directive might be used to map file systems that lie outside documentroot to the web space.

Use the following procedure to configure the bypass of Oracle Business Intelligence static files from the WebLogic Server and directing Oracle HTTP Server to serve the static file requests.

To bypass static files when using Oracle HTTP Server:

1. Open the httpd.conf file for editing.

   This file is located in *ORACLE_HOME*/Apache/Apache/conf.

2. Verify that the following directive is not commented out in the httpd.conf file:

```
LoadModule expires_module modules/mod_expires.so
```

Note: For Apache versions prior to 1.3.15 on Windows, the directive is LoadModule expires_module modules/ApacheModuleExpires.dll

3. Add the configuration into the httpd.conf file, as shown in the following example:

```
<Directory $ORACLE_HOME/bifoundation/web/app/res>
Order allow,deny
Allow from all
</Directory>
```

```
<Directory $ORACLE_HOME/bifoundation/web/app/olh>
Order allow,deny
Allow from all
</Directory>
Alias /OBIContent_res $ORACLE_HOME/bifoundation/web/app/res
Alias /OBIContent_olh $ORACLE_HOME/bifoundation/web/app/olh

<VirtualHost *:*>
#ServerName bi.example.com
RewriteEngine on
RewriteRule ^/analytics/res/(.*)$ /OBIContent_res/$1 [PT]
RewriteRule ^/analytics/olh/(.*)$ /OBIContent_olh/$1 [PT]
</VirtualHost>
```

where:

The aliases OBIContent_res and OBIContent_olh are used to map the static files located in the following directories on the application server:

– *ORACLE_HOME*/bifoundation/web/app/res

– *ORACLE_HOME*/bifoundation/web/app/olh

> **Note:** Replace bi.example.com with the virtual host name for your deployment.

4. If SSL has been enabled on the HTTP Server, then add the following lines to the ssl.conf file that is located in the same directory as httpd.conf:

```
RewriteEngine on
RewriteRule ^/analytics/res/(.*)$ /OBIContent_res/$1 [PT]
RewriteRule ^/analytics/olh/(.*)$ /OBIContent_olh/$1 [PT]
As shown below:
<VirtualHost _default_:443>
# General setup for the virtual host
DocumentRoot "C:\OAS10.1.3\OracleAS_1\Apache\Apache\htdocs"
ServerName bi.example.com

RewriteEngine on
RewriteRule ^/analytics/res/(.*)$ /OBIContent_res/$1 [PT]
RewriteRule ^/analytics/olh/(.*)$ /OBIContent_olh/$1 [PT]

ServerAdmin you@your.address
```

5. Restart Oracle HTTP Server.

Part IV

Resolving Issues

This part explains how to resolve issues in Oracle Business Intelligence. It contains the following chapters:

- Chapter 8, "Diagnosing and Resolving Issues in Oracle Business Intelligence"
- Chapter 9, "Managing Usage Tracking"

8

Diagnosing and Resolving Issues in Oracle Business Intelligence

This chapter describes how to diagnose and resolve issues. Oracle Business Intelligence provides easy ways to find the causes and solutions to issues. You can do the following to resolve issues:

- View recent issues with the system on the Overview page in Fusion Middleware Control.
- Drill into the problems using the diagnostics pages in Fusion Middleware Control.
- Track issues across components (for example, regarding performance or availability).
- View the log files from within Fusion Middleware Control.
- Resolve the issues by changing configuration of the Oracle Business Intelligence system.

The principal activities that you perform to support issue resolution include:

- Examination and configuration of error and diagnostic log information. For information, see:
  - Section 8.1, "Viewing And Configuring Diagnostic Log Files"
  - Section 8.2, "Understanding Diagnostic Log and Log Configuration Files"
  - Section 8.3, "Managing the Query Log"
  - Section 8.4, "Logging in Oracle BI Presentation Services"
- Examination of system and process metrics to understand availability and performance issues. For information, see:
  - Section 7, "Managing Performance Tuning and Query Caching"

8.1 Viewing And Configuring Diagnostic Log Files

You can view diagnostic log files and configure settings that affect diagnostic log files and the information that is written to them, as described in the following sections:

- Section 8.1.1, "Using Fusion Middleware Control to View Log Information, Error Messages, and Alerts"
- Section 8.1.2, "Configuring Log File Rotation Policy and Specifying Log Levels"

8.1.1 Using Fusion Middleware Control to View Log Information, Error Messages, and Alerts

You can search for and view the log entries for Oracle Business Intelligence components using Fusion Middleware Control Log Viewer. The log files can be searched for log messages, and you can apply filters that can, for example, target a certain date range, user, user transaction, or level of message (error, warning, notification, and so on). You can also view log files in their entirety from the Fusion Middleware Control Log Viewer.

When log entries for error messages and warnings are generated across multiple log files, they can be difficult to trace. However, it is possible to view logging information for specific user transactions across multiple log files. Transaction level logging associates a unique transaction ID, which is called the Execution Context ID (ECID), with every log and error message that is generated in response to a user request. This logging enables rapid diagnosis of the cause of underlying issues. However, some messages in the log (for example system messages for server startup or shutdown) do not have a transactional attribute. All log messages that are related to client requests do have a transactional attribute.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to view log information, error messages, and alerts:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Diagnostics page.

   Click the **Help** button on the page to access the page-level help for its elements.
3. View lists of the following:
   - recent errors under the **Most Recent Errors** region
   - recent warnings under the **Most Recent Warnings** region
4. Select a link under View/Search All Log Files and View/Search Log Files By Component to display messages for all log files, or for the messages for the log files of a specified component. Click the **Help** button on the page to access the page-level help for the following links:
   - **Search the log files using the Log Viewer**
   - **Presentation Services Log**
   - **Server Log**
   - **Scheduler Log**
   - **JavaHost Log**
   - **Cluster Controller Log**
   - **Action Services Log**
   - **Security Services Log**
   - **Administrator Services Log**

Fusion Middleware Control displays messages in the Log Messages page that correspond to your selection.

5. Enter appropriate search criteria to display corresponding error messages.

   To view messages by ECID, click **View Related Messages** and select the **by ECID (Execution Context ID)** menu option.

6. Select one or more rows to view the log file entry details for the selected messages.

   For more information about the elements that are displayed in the Log Viewer, see the Fusion Middleware help.

8.1.2 Configuring Log File Rotation Policy and Specifying Log Levels

You can configure criteria that determine when a new log file must be created, based on the size of the log file and the age of the log file. You can also specify log levels to determine what level of message is written to the log files.

This section contains the following topics:

- Section 8.1.2.1, "Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels"
- Section 8.1.2.2, "Manually Changing Additional Log File Settings"

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

8.1.2.1 Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to configure log file rotation policy and specify log levels:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Diagnostics page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page.

   For example, you can specify which log levels to use, and for some you can set their granularity.

   Click the **Help** button on the page to access the page-level help for the following options:

   Log Configuration

   - **Maximum File Size** option
   - **Maximum Log Age** option

   Query Logs

   - **Maximum File Size** option

- **Maximum Log Age** option

Log Levels

- **Incident Error** option
- **Error** option
- **Warning** option
- **Notification** option
- **Trace** option

5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

8.1.2.2 Manually Changing Additional Log File Settings

In addition to the log file settings that you can change in Fusion Middleware Control, you can change other settings manually. Use various elements in the log configuration file for a component to change these settings.

> **Note:** Editing a diagnostic log configuration file for a single component is not advised, because changes might subsequently be overwritten. For information, see Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings."

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually change the settings that configure the log file format:

1. Open the component log configuration file as described in Section 8.2.2, "What Are Diagnostic Log Configuration Files and Where are They Located?"
2. Locate the section in which you must add the Format element, which specifies the log file format. The default is ODL-TEXT.

   To use the Fusion Middleware Control Log Viewer to view and search through the log files for Oracle Business Intelligence, then the files must be in either ODL-Text or ODL-XML format.
3. Include the element and its ancestor elements as appropriate, as shown in the following example:

   ```
   <server>
     <ServerInstance>
       <Log>
         <Format>ODL-TEXT</Format>
       </Log>
     </ServerInstance>
   </server>
   ```

   For an example of a JavaHost Server diagnostic log configuration file, see Example 8–2.
4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

8.2 Understanding Diagnostic Log and Log Configuration Files

This section discusses diagnostic log files and diagnostic log configuration files, and contains the following topics:

- Section 8.2.1, "What Are Diagnostic Log Files and Where are They Located?"
- Section 8.2.2, "What Are Diagnostic Log Configuration Files and Where are They Located?"
- Section 8.2.3, "What Are Log File Message Categories and Levels?"
- Section 8.2.4, "What is Log File Rotation?"

8.2.1 What Are Diagnostic Log Files and Where are They Located?

Diagnostic log files are files used to store message information that is generated by Oracle Business Intelligence servers. These log files are stored in the following location:

ORACLE_INSTANCE\diagnostics\logs\component_type\coreapplication

The following diagnostic log files are used in Oracle Business Intelligence:

- Presentation Services
  - \CatalogCrawler\sawcatalogcrawlerlogsys*n*.log — The catalog crawler log file, which is not searchable in the Fusion Middleware Control Log Viewer.
  - sawlog*n*.log — The Presentation Services log file that represents the latest part of diagnostic log output and is searchable in the Fusion Middleware Control Log Viewer.

  For more information specifically about Presentation Services logging, see Section 8.4, "Logging in Oracle BI Presentation Services."
- Oracle BI Server
  - nqquery<n>.log — The Oracle BI Server query log, which is not searchable in the Fusion Middleware Control Log Viewer.

    <*n*> = date and timestamp, for example nqquery-20101209-2135.log
  - nqserver<n>.log — The Oracle BI Server main diagnostic log, which is searchable in the Fusion Middleware Control Log Viewer.

    <*n*> = date and timestamp, for example nqserver-20101209-2135.log
  - nqsadmintool.log — The log for the Oracle BI Administration Tool.
  - Oracle BI Server utilities — For example, biserverxmlexec and equalizerpds, also generate their own logs when they are run.
- JavaHost
  - jh-*n*.log — The JavaHost Server main diagnostic log,which is searchable in the Fusion Middleware Control Log Viewer.

    <*n*> = date and timestamp, for example jh-20100909-2135.log
- Oracle BI Scheduler
  - nqscheduler-<*n*>.log — The Oracle BI Scheduler log file, which is searchable in the Fusion Middleware Control Log Viewer.

    <*n*> = date and timestamp, for example nqscheduler-20100909-2135.log

- Cluster Controller
  - nqcluster-yyyyMMdd-hhmm.log — The Oracle BI Cluster Controller diagnostic file, which is searchable in the Fusion Middleware Control Log Viewer.

    <*n*> = date and timestamp, for example nqcluster-20100909-2135.log
- BI JEE log (Action Services and Security Services), both of the following log files are searchable in the Fusion Middleware Control Log Viewer:
  - AdminServer-diagnostic.log
  - bi_server1-diagnostic.log
- Upgrade

  Log files for the upgrade of Oracle Business Intelligence are created in the following location:

  ORACLE_HOME\upgrade\logs

  For information about upgrade log files, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*. These files are not searchable in the Fusion Middleware Control Log Viewer.

8.2.2 What Are Diagnostic Log Configuration Files and Where are They Located?

Diagnostic log configuration files control output to diagnostic log files for Oracle Business Intelligence.

> **Note:** Editing a diagnostic log configuration file for a single component is not advised, because changes might subsequently be overwritten. For information, see Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings."

Log configuration files for Oracle Business Intelligence are stored in the following locations:

ORACLE_INSTANCE\config*component_type**bi_component_name*

For example:

- \OPMN\opmn\opmn.xml
- \OracleBIClusterControllerComponent\coreapplication_obiccs1\ccslogconfig.xml
- \OracleBIJavaHostComponent\coreapplication_obijh1\logging_config.xml
- \OracleBIPresentationServicesComponent\coreapplication_obips1\instanceconfig.xml
- \OracleBISchedulerComponent\coreapplication_obisch1\instanceconfig.xml
- \OracleBIServerComponent\coreapplication_obis1\logconfig.xml

About Formats in Diagnostic Log Configuration Files

Diagnostic log configuration files conform to the Oracle Diagnostic Log (ODL) standard, although they can differ slightly in appearance.

Example 8–1 and Example 8–2 illustrate two of the log configuration files for Oracle Business Intelligence.

Example 8–1 BI Server Diagnostic Log Configuration File Format

```
<server>
  <ServerInstance>
    <Log>
      <MaximumFileSizeKb>10000</MaximumFileSizeKb>
      <MaximumLogAgeDay>60</MaximumLogAgeDay>
      <Format>ODL-TEXT</Format>
      <Level>
        <IncidentError>1</IncidentError>
        <Error>1</Error>
        <Warning>16</Warning>
        <Notification>1</Notification>
        <Trace>16</Trace>
      </Level>
    </Log>
    <UserLog>
      <MaximumFileSizeKb>10000</MaximumFileSizeKb>
      <MaximumLogAgeDay>10</MaximumLogAgeDay>
      <Format>ODL-TEXT</Format>
    </UserLog>
  </ServerInstance>
</server>
```

Example 8–2 JavaHost Server Diagnostic Log Configuration File Format

```
<?xml version = '1.0' encoding = 'utf-8'?>
<logging_configuration>
   <log_handlers>
      <log_handler name='odl-handler'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
         <property name='path' value='C:\oracle_bi_ee_
BIFNDNPTPSNT0911060426S-Release\jhlogs\javahost.log'/>
         <property name='maxFileSize' value='1000000'/>
         <property name='maxLogSize' value='5000000'/>
      </log_handler>
   </log_handlers>
   <loggers>
      <logger name='saw' level='NOTIFICATION:1' useParentHandlers='false'>
         <handler name='odl-handler'/>
      </logger>
   </loggers>
</logging_configuration>
```

Oracle Business Intelligence components control their diagnostic log files by using server-specific settings in their log configuration files, for example:

- Oracle BI Presentation Services log configuration file:

  - `writerClassId` settings configure messages that are written to the sawlog.log file.

- Oracle BI Server log configuration file:

  - `Log` settings configure messages that are written to the nqserver.log file.

  - `UserLog` settings configure messages that are written to the nqquery.log file.

- Oracle BI Scheduler log configuration file:

  - `Log` settings configure messages that are written to the nqscheduler.log file.

- JavaHost Server log configuration file:

- `log_handlers` elements and sub-elements enable configuration of the log file rotation policy and the specification of the log file name and its location.

- `loggers` elements and sub-elements enable appropriate handling of Java component (JavaHost Server) log levels, by mapping the JavaHost Server log levels to the standard Oracle Diagnostic Log (ODL) log levels.

8.2.3 What Are Log File Message Categories and Levels?

Categories and levels for log file messages define the detail and level of importance with which messages are written to a log file. Fusion Middleware Control enables you to control these settings in the logconfig.xml file.

Each message category in a log file for Oracle Business Intelligence is set to a specific default value between 1 and 32, and only messages with a level less than or equal to the log level is logged.

Log file message categories are described in Table 8–1.

Table 8–1 Log File Message Category Levels

Category:Level	Description
IncidentError:1	A serious problem caused by unknown reasons has occurred. You can fix the problem only by contacting Oracle Support Services. No performance impact.
Error:1	A problem that requires attention from the system administrator has occurred. No performance impact.
Warning:1	An action occurred or a condition was discovered that should be reviewed and might require action before an error occurs. No performance impact.
Notification:1	A report of a normal action or event has occurred. This could be a user operation, such as "login completed" or an automatic operation such as a log file rotation. No performance impact.
Notification:16	A configuration-related message or problem has occurred. Low performance impact. It should be possible to enable this level broadly in a production environment without having a significant performance impact in the software.
Trace:1	A trace or debug message that is used for debugging or performance monitoring has been written. Typically this message contains detailed event data that is clear enough to be understood by someone who does not know internal implementation details. Small performance impact. This level might be enabled broadly occasionally on a production environment to debug issues with the software. Enabling logging at this level might have a small performance impact, but not to the point of making the software unusable.
Trace:16	A fairly detailed trace or debug message has been written. The message is clear enough to be understood by Oracle Support engineers who have a deep knowledge of the product but might not know full details of the internal implementation. High performance impact. This level should not be enabled on a production environment, except on special situations to debug issues with the software.

Table 8–1 (Cont.) Log File Message Category Levels

Category:Level	Description
Trace:32	A highly detailed trace or debug message has been written. The message is intended for an Oracle developer working on the software who knows enough details about the implementation of the subsystem that generates the message. Very high performance impact. This level is not expected to be enabled in a production environment and should be used only to debug the software on a test or development environment.

In the following log configuration file example, in the Notification message category, only level 1 messages are logged. If the log level is set to 0, then nothing is logged for that message category.

```
<Level>
  <IncidentError>1</IncidentError>
  <Error>1</Error>
  <Warning>1</Warning>
  <Notification>1</Notification>
  <Trace>1</Trace>
</Level>
```

Avoid manually changing the default settings in the log file. Use Fusion Middleware Control to make changes. For more information, see Section 8.1.2.1, "Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels".

8.2.4 What is Log File Rotation?

Log file rotation is the creation of new log files, when the file exceeds a specified threshold or date. Take the MaximumFileSizeKb setting for the component log configuration file for the Oracle BI Scheduler as an example. Whenever a log file exceeds the size that is specified by this setting, then the existing Scheduler log file is renamed, and a new log file is created. Additionally, a log file date that is older than the MaximumLogAgeDay setting is deleted. The file naming convention for the Scheduler is as follows:

- nqscheduler.log — The latest log file.
- nqscheduler-<*n*>.log — The renamed previous log file.

  where <*n*> = date and timestamp, for example nqscheduler-20100909-2135.log

The naming convention that is used for settings in log configuration files differs slightly across components.

For more information, see Section 8.1.2.1, "Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels."

8.3 Managing the Query Log

The Oracle BI Server provides a facility for logging query activity at the individual user level. Use logging for quality assurance testing, debugging, and troubleshooting by Oracle Support. In production mode, query logging is typically disabled.

The query log file is named nqquery.log, and is located in:

ORACLE_INSTANCE\diagnostics\logs\*component_type**bi_component_name*

Oracle BI Server query logging is tracked at a user level. It is a resource-intensive process if you track the entire user community.

> **Note:** For production systems, it is recommended that query logging be enabled only for a very targeted user community. In production systems, you can use usage tracking as the production-level logging facility. See Chapter 9, "Managing Usage Tracking" for more information.

It is recommended that you only test users when the user name clearly indicates it is a test user and have verified that query logging is enabled. If logging is enabled for such users, then it is recommended that they be given names such as sales_admin_with_logging, sales_dev_with_logging, or sales_test_with_logging, so that you can readily identify them. Even production administrator logins should not have query logging enabled, because it could strain the available resources.

You should also disable query logging for the following:

- The SQL statement in the initialization string. The **Initialization string** field is in the Initialization Block dialog box, in the General tab.

  The LOGGING column references stored values for the log level.

- The logging level should be set to 0 (zero) for each production user. The **Logging level** field is in the User dialog box, in the User tab. In the Administration Tool, select **Identity** from the Manage option on the main toolbar. In the Security Manager dialog, double-click a user and select the User tab.

This section contains the following topics:

- Section 8.3.1, "Configuring Query Logging"
- Section 8.3.2, "Using the Log Viewer"

8.3.1 Configuring Query Logging

This section includes information about setting the size of the query log, choosing a logging level, and enabling query logging for a user.

Because query logging can produce very large log files, the logging system is turned off by default. You can enable logging to test that the repository is configured properly, to monitor activity on the system, to help solve performance problems, or to assist Oracle Support Services. You must enable query logging on the system for each user whose queries you want logged. You do this using the Oracle BI Administration Tool.

8.3.1.1 Setting the Query Logging Level

You can enable query logging levels for individual users, as described in Section 8.3.1.2, "Setting the Query Logging Level for a User." You cannot configure a logging level for a group.

A session variable overrides the logging level for a particular user. For example, if the administrator has a logging level of 4 and the session variable logging level is defined as the default 0 (zero) in the repository, then the logging level for the administrator is 0.

Set the logging level based on the amount of logging that you want to do. In normal operations, logging is generally disabled (that is, the logging level is set to 0). If you decide to enable logging, then choose a logging level of 1 or 2. These two levels are designed for use by administrators.

You might want to diagnose performance or data issues by setting a temporary log level for a query. You can enable query logging for a select statement by adding a prefix clause in the Advanced SQL Clauses section of the Advanced tab in Oracle BI Presentation Services. For example, for the select statement:

```
SELECT year, product, sum(revenue) FROM time, products, facts;
```

You can specify the logging level of 5 in the **Prefix** field as follows:

```
Set Variable LOGLEVEL=5;
```

For this query, the logging level of 5 is used regardless of the value of the underlying `LOGLEVEL` variable.

> **Note:** Use logging levels greater than 2 only with the assistance of Oracle Support Services.

The query logging levels are described in Table 8–2.

Table 8–2 Query Logging Levels

Logging Level	Information That Is Logged
Level 0	No logging.
Level 1	Logs the SQL statement issued from the client application. Also logs the following: ■ Physical Query Response Time — The time for a query to be processed in the back-end database. ■ Number of physical queries — The number of queries that are processed by the back-end database. ■ Cumulative time — The sum of time for all physical queries for a request (that is, the sum of all back-end database processing times and DB-connect times). ■ DB-Connect time — The time taken to connect to the back-end database. ■ Query cache processing — The time taken to process the logical query from the cache. ■ Elapsed time — The time that has elapsed from when the logical query is presented to the BI Server until the result is returned to the user. Elapsed time can never be less than response time, because elapsed time takes into account the small extra time between the logical query being presented to the BI Server to the start of preparation of the query. In cases where this delta time is negligible, the elapsed time equals the response time. ■ Response time — The time taken for the logical query to prepare, execute, and fetch the last record. This matches the TOTAL_TIME_SEC that is logged in usage tracking, as described in Section 9.2, "Description of the Usage Tracking Data." ■ Compilation time — The time taken to compile the logical query. ■ For each query, logs the query status (success, failure, termination, or timeout), and the user ID, session ID, and request ID.

Table 8–2 (Cont.) Query Logging Levels

Logging Level	Information That Is Logged
Level 2	Logs everything logged in Level 1. Additionally, for each query, logs the repository name, business model name, subject area name, SQL statement issued against the physical database, queries issued against the cache, number of rows returned from each query against a physical database and from queries issued against the cache, and the number of rows returned to the client application.
Level 3	Logs everything logged in Level 2. Additionally, adds a log entry for the logical query plan, when a query that was supposed to seed the cache was not inserted into the cache, when existing cache entries are purged to make room for the current query, and when the attempt to update the exact match hit detector fails. Do not select this level without the assistance of Oracle Support Services.
Level 4	Logs everything logged in Level 3. Additionally, logs the query execution plan. Do not select this level without the assistance of Oracle Support Services.
Level 5	Logs everything logged in Level 4. Additionally, logs intermediate row counts at various points in the execution plan. Do not select this level without the assistance of Oracle Support Services.
Level 6 and 7	Not used.

8.3.1.2 Setting the Query Logging Level for a User

To set the query logging level for a user:

1. In the Oracle BI Administration Tool, select **Manage**, then **Identity**.

   The Security Manager dialog box is displayed.

2. Double-click the name of the user for which you want to set the query logging level.

   The User dialog box is displayed.

3. Set the logging level by clicking the **Up** or **Down** arrows next to the **Logging Level** field.

   To disable query logging for a user, set the logging level to 0.

4. Click **OK**.

8.3.2 Using the Log Viewer

Use the Oracle Business Intelligence Log Viewer utility (or a text editor) to view the query log. Each entry in the query log is tagged with the name of the user who issued the query, the session ID of the session in which the query was initiated, and the request ID of the individual query.

To run the Log Viewer utility (which is located on Windows in \MW_HOME\ORACLE_HOME\bifoundation\server\bin\nqlogviewer.exe), open a command prompt, and enter `nqlogviewer` with any combination of its arguments. The syntax is as follows:

```
nqlogviewer [-u user_name] [-f log_input_filename]
          [-o output_result_filename]
          [-s session_ID] [-r request_ID]
```

where:

- `user_name` is the name of a user in the Oracle Business Intelligence repository. This parameter limits the scope to entries for a particular user. If not specified, all users for whom query logging is enabled are displayed.
- `log_input_filename` is the name of an existing log file from where the content is taken. This parameter is required.
- `output_result_filename` is the name of a file in which to store the output of the log. If the file exists, then the results are appended to the file. If the file does not exist, then a new file is created. If this argument is not specified, then output is sent to the monitor screen.
- `session_ID` is the session ID of the user session. The BI Server assigns each session a unique ID when the session is initiated. This parameter limits the scope of the log entries to the specified session ID. If not specified, then all session IDs are displayed.
- `request_ID` is the request ID of an individual query. The BI Server assigns each query a unique ID when the query is initiated. This parameter limits the scope of the log entries to the specified request ID. If not specified, then all request IDs are displayed.

  The request ID is unique among the active requests, but not necessarily unique during the session. Request IDs are generated in a circular manner, and if a request is closed or if the session is long enough, then a request ID is reused.

You can also locate user names, session IDs, and request IDs through the Session Manager. See *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for information.

Administrators can view the query log using the **Manage Sessions** option in the Presentation Services Administration page.

8.3.2.1 Interpreting the Log Records

After you have logged some query information and started the log viewer, you can analyze the log. Log entries for levels 1 and 2 are generally self-explanatory. The log entries can provide insights to help DBAs in charge of the underlying databases tune them for optimum query performance. The query log can also help you check the accuracy of applications that use the BI Server.

The log is divided into the following sections:

- **SQL Request —** This section lists the SQL statement that is issued from the client application. You can use this information to rerun the query from the same application, or from a different application.
- **General Query Information —** This section lists the repository, the business model, and the subject area from which the query was run. You can use this information to provide statistics on query usage that you can use to set priorities for future application development and system management.
- **Database Query —** This section begins with an entry that reads "Sending query to the database named <*data_source_name*>," where <*data_source_name*> is the name of the data source to which the BI Server is connecting. Multiple database queries can be sent to one or more data sources. Each query has an entry in the log.

  The database query section has several uses, such as recording the SQL statement that was sent to the underlying databases. You can use this logged SQL statement to run queries directly against the database for performance tuning, results verification, or other testing purposes. You can also use this information to examine the tables that are being queried to verify that aggregate navigation is working as you expect. If you understand the structure of the underlying database, then it might also provide some insights into potential performance improvements, such as useful aggregate tables or indexes to build.

- **Query Status** — The query success entry in the log indicates whether the query completed successfully, or failed. You can search through the log for failed queries to determine why they failed. For example, all the queries during a particular time period might have failed due to database downtime.

8.4 Logging in Oracle BI Presentation Services

This section describes logging specifically in Presentation Services and contains the following topics:

- Section 8.4.1, "Using the Oracle BI Presentation Services Logging Facility"
- Section 8.4.2, "Structure for the Oracle BI Presentation Services Configuration File"
- Section 8.4.3, "Oracle BI Presentation Services Message Structure"
- Section 8.4.4, "Oracle BI Presentation Services Log Filters"
- Section 8.4.5, "Diagnosing Issues with the Agents"

For general information about logging in Oracle Business Intelligence, see Section 8.2, "Understanding Diagnostic Log and Log Configuration Files."

8.4.1 Using the Oracle BI Presentation Services Logging Facility

By default, Oracle BI Presentation Services is configured to log all error events and informational and warning events of sufficient importance. An example of an important informational event is a server starting up or a server shutting down. Log files are named sawlog*xx*.log, where the *xx* is replaced by an incremented number.

To debug specific issues that a user might be encountering, the logging level can be increased to log more information than the default configuration. For example, while debugging a particular Oracle BI Presentation Services connectivity issue, you can increase the maximum logging on the saw.odbc log source only. This adds detailed logging for that component, without cluttering the log with detailed logging from other events. All Oracle BI Presentation Services configuration information is loaded from the instanceconfig.xml file.

Caution: Because logging affects performance, do not increase the logging on a production implementation, except to diagnose specific issues.

8.4.2 Structure for the Oracle BI Presentation Services Configuration File

The structure of the configuration file is shown in Example 8–3. The cardinality of each node is shown in brackets.

Example 8–3 Structure of Log Section in instanceconfig.xml File

```
Logging [1..1]
Writers [0..1]
Writer [0..1]
```

```
WriterClassGroups [0..1]
Filters [0..1]
FilterRecord [0..n]
```

An example of an instanceconfig.xml file that has four writers is shown in Example 8–4.

Example 8–4 instanceconfig.xml File with Four Writers

```
<?xml version="1.0" ?>
<Server>
. . . . . . .
<Logging>
<Writers>
<Writer implementation="FileLogWriter" name="Global File Logger"
writerClassId="1" dir="{%ORACLE_BIPS_INSTANCE_LOGDIR%}" filePrefix="sawlog"
maxFileSizeKb="10000" filesN="10" fmtName="ODL-Text" ODLLogFilePath="{%ORACLE_
BIPS_INSTANCE_LOGDIR%}/diagnostic.log"/>
<Writer implementation="CoutWriter" name="Global Output Logger"
writerClassId="2" />
<Writer implementation="EventLogWriter" name="Event Logger"
writerClassId="3" />
<Writer implementation="CrashWriter" name="CrashWriter"
writerClassId="4"
/>
</Writers>
<WriterClassGroups>
<WriterClassGroup name="All">1,2,3,4</WriterClassGroup>
<WriterClassGroup name="File">1</WriterClassGroup>
<WriterClassGroup name="Console">2</WriterClassGroup>
<WriterClassGroup name="EventLog">3</WriterClassGroup>
<WriterClassGroup name="Crash">4</WriterClassGroup>
</WriterClassGroups>
<Filters>
<FilterRecord writerClassGroup="Console" path = "saw" information="1" warning="31"
error="31" trace="0" incident_error="32" />
<FilterRecord writerClassGroup="File" path = "saw" information="1" warning="31"
error="31" trace="0" incident_error="32" />
<FilterRecord writerClassGroup="File" path="saw.mktgsqlsubsystem.joblog"
information="1" warning="2" error="31" trace="0" incident_error="32"/>
<FilterRecord writerClassGroup="File" path="saw.httpserver.request"
information="16" warning="32" error="32" trace="0" incident_error="32"/>
<FilterRecord writerClassGroup="File" path="saw.httpserver.response"
information="16" warning="32" error="32" trace="0" incident_error="32"/>
</Filters>
</Logging>
</Server>
```

Table 8–3 contains a description of each node in the configuration hierarchy.

Table 8–3 Oracle BI Presentation Services Log Configuration File Elements

Element	Attribute	Description
Writers	None	Contains writers configuration. This configuration is loaded on startup.
Writer	None	Configures a writer.

Table 8–3 (Cont.) Oracle BI Presentation Services Log Configuration File Elements

Element	Attribute	Description
Writer	disableCentralControl	(Optional) Determines that this entry is not updated by Fusion Middleware Control. Default value is true.
Writer	implementation	The following implementations are defined: - **FileLogWriter.** Writes to a disk file. - **CoutWriter.** Writes to standard output. - **EventLogWriter.** Writes to a Windows event log or UNIX syslog. - **CrashWriter.** A Windows only facility that writes to a crash dump file when Presentation Services attempts to log from a specific source file and line number. Used in a production environment for information of some loggable but irrecoverable error (for example, failed NQTEST). **Note:** Use this implementation with care as it might leave the server in an unstable state. Use this implementation in very rare diagnostic-only scenarios on a test system. On Windows, CrashWriter requires the appropriate version of `dbghelp.dll` (at least 6.0.17.0). The correct `dbghelp.dll` can be found in `support/windows/system32`. Put this DLL in the `WINNT/system32` or in the `main/bin` directory. No registration is required.
Writer	name	Unique name for the writer.
Writer	writerClassId	Specifies an nteger number in the range 1 through 10. This number is used by filters to allow or prohibit logging. Each distinct writer must have a unique value, which is used later for filter configuration. Different writers might have the same class ID, but if they do, those writers cannot be distinguished by filters.
Writer	fmtName	(Optional) Specifies the format of logged messages. Valid values are: - **default - 10g style**. Formats messages with identifying headings. - **ODL-TEXT**. Formats messages in Oracle Diagnostic Text format. - **ODL-XML**. Formats messages in Oracle Diagnostic XML format. If you do not set this attribute, then logged messages are displayed in the default format which for file log writers is 10g style and for console is ODL-TEXT. See Section 8.4.2.1, "Examples of the Formats of Logged Messages" for examples.

Table 8–3 (Cont.) Oracle BI Presentation Services Log Configuration File Elements

Element	Attribute	Description
Writer (FileLogWriter specific attribute)	dir	Specifies the directory where log files are created.
Writer (FileLogWriter specific attribute)	ODLLogFilePath	Specifies the file that Fusion Middleware Control displays in the Log Viewer.
Writer (FileLogWriter specific attribute)	maxFileSizeKb	Specifies the maximum size of the logging file in kilobytes. When the file size limit is reached, the file is closed and a new logging file is created.
Writer (FileLogWriter specific attribute)	filePrefix	Specifies the prefix for log files.
Writer (FileLogWriter specific attribute)	filesN	Specifies the maximum number of logging files. When this number is exceeded, the first file is deleted and re-created again. Then the logger starts to write to the beginning of the first file.
Writer (EventLogWriter specific attribute)	winSource	Specifies the event log source for logged events.
Writer (CrashWriter specific attribute)	file	Specifies the dump file path. On Windows, a dump file is created in `bin/coredumps` and Presentation Services continues to run.
Writer (CrashWriter specific attribute)	line	Dump file line number.
WriterClassGroups	None	Contains the definition for writer classes. A writer class is a group of writer class IDs.
WriterClassGroup (Contains [as child text] a comma-delimited list of class IDs.)	name	Specifies the name of the WriterClassGroup.
Filters	None	Contains filter configuration.
FilterRecord	writerClassGroup	Specifies the group of writers to which this record is applied. WriterClassGroup should be defined previously in the WriterClassGroups section.
FilterRecord	disableCentralControl	(Optional) Determines that this entry is not updated by Fusion Middleware Control. Default value is true.
FilterRecord	path	Specifies the log source path. To enable the logging of SOAP information, enter the following value: saw.httpserver.request.soaprequest The current filter record is applied to the software component that is identified by that path and all its subcomponents.
FilterRecord	information	Contains an integer that specifies the severity of the corresponding message type. Only messages with a severity index less than the provided number are logged.

Table 8–3 (Cont.) Oracle BI Presentation Services Log Configuration File Elements

Element	Attribute	Description
FilterRecord	warning	Contains an integer that specifies the severity of the corresponding message type. Only messages with a severity index less than the provided number are logged.
FilterRecord	error	Contains an integer that specifies the severity of the corresponding message type. Only messages with a severity index less than the provided number are logged.
FilterRecord	trace	Contains an integer that specifies the severity of the corresponding message type. Only messages with a severity index less than the provided number are logged.
FilterRecord	incident_error	Contains an integer that specifies the severity of the corresponding message type. Only messages with a severity index less than the provided number are logged.

8.4.2.1 Examples of the Formats of Logged Messages

The fmtName attribute of the Writer element formats logged messages in one of three formats: default (10g style), ODL-TEXT, and ODL-XML. The following entries are examples of these formats.

Example 8–5 Default Format

The default format generates messages with identifying headings, such as:

```
Type: Information
Severity: 30
Time: Wed Jul 26 11:22:20 2006
File: project\sawserver\sawserver.cpp
Line: 399
Properties: ThreadID-2552
Location:
             saw.sawserver
             saw.sawserver.initializesawserver
             saw.sawserver
Oracle BI Presentation Services has started successfully.
```

Example 8–6 ODL-TEXT Format

The short format generates messages in a shortened form without identifying headings, such as:

```
[timestamp] [component id] [messagetype:level] [message-id] [module id]
([field-name: field-value])* message-text [[
supplemental-detail
]]

[2010-05-27T10:51:20.000-07:00] [OBIPS] [NOTIFICATION:1] [] [saw.sawserver] [ecid:
1243446680218334471555761] [tid: 2552] Oracle BI Presentation Services (OBIPS)
11.1.1.2 (Build 0) are starting up.[[
File:sawserver.cpp
Line:432
Location:
    saw.sawserver
    saw.sawserver.initializesawserver
    saw.sawserver
ecid: 12434466802183344715557 61
]]
```

Example 8–7 ODL-XML Format

The xml format generates messages in XML format, such as:

```
<msg time="2010-05-08T18:41:05.000+00:00"
comp_id="OBIPS" type="NOTIFICATION" level="1" msg_id=""
module="saw.sawserver" ecid="12418044651787424262 8761" tid="127c">
<txt> Oracle BI Presentation Services has started successfully</txt>
<suppl_detail />
</msg>
```

8.4.3 Oracle BI Presentation Services Message Structure

Each message that is logged by Presentation Services has several components, as described in Table 8–4.

Table 8–4 Components of Presentation Services Log Message

Message Component	Description
Message Text	The text of the log message to the user.
Message Type	One of five types: information, warning, error, incident_error or trace. For information, see Table 8–1, " Log File Message Category Levels".
Severity	The severity is represented as a positive integer. The lower the value, the more important the message. A message with severity of 0 is the most important type of message, while a message with a severity of 32 is not important at all.
Message Properties	Properties indicate other kinds of information. The kind varies among messages and might include user name, the IP address of the client browser, the thread ID, and so on.

8.4.4 Oracle BI Presentation Services Log Filters

FilterRecords customize logging details. Use FilterRecords to specify the implementation (output type) and logging levels for categories of Web logs: Incident Error, Error, Trace, Warnings, and Information.

In the following example, the first two FilterRecords contain the following string:

```
path="saw"
```

This string logs the informational events at level 1, the error messages at level 31, and so on:

```
<FilterRecord writerClassGroup="Console" path="saw" information="1" warning="31"
error="31" trace="0" incident_error="32" />
<FilterRecord writerClassGroup="File" path="saw" information="1" warning="31"
error="31" trace="0" incident_error="32" />
<FilterRecord writerClassGroup="File" path="saw.mktgsqlsubsystem.joblog"
information="1" warning="2" error="31" trace="0" incident_error="32"/>
```

This high-level path applies to every event.

You can customize FilterRecords by adding new FilterRecords, such as the third one shown in the preceding example, with finer-grain specification of log levels for events of various types. In this example, information is being logged to a disk file from saw.mktgsqlsubsystem.log, which generates Marketing job events.

You can disable logging of job details by changing the information level from 1 to 0, as shown in the following example, or by commenting out the lines:

```
<FilterRecord writerClassGroup="Console" path="saw" information="1" warning="31"
error="31" trace="0" incident_error="32" />
<FilterRecord writerClassGroup="File" path="saw" information="1" warning="31"
error="31" trace="0" incident_error="32" />
<FilterRecord writerClassGroup="File" path="saw.mktgsqlsubsystem.joblog"
information="1" warning="2" error="31" trace="0" incident_error="32"/>
```

8.4.5 Diagnosing Issues with the Agents

If an agent fails to execute fully or if debugging is turned on in Oracle BI Scheduler, then a log file is generated for the agent.

You manually turn on debugging by setting the Debug element to True in the Oracle BI Scheduler instanceconfig.xml file. (For information, see Section 8.2.2, "What Are Diagnostic Log Configuration Files and Where are They Located?")

The location for agent log files is specified in the instanceconfig.xml file for the Oracle BI Scheduler. (For information, see Section 20.3.3.3, "Agent Scheduler Configuration Settings.") The default location for log files is the Log directory in the Oracle Business Intelligence installation directory on the computer where the Oracle BI Scheduler is installed.

The log file name has the following format:

Agent-*JobID*-*InstanceID*.*xxx*

where:

- Agent is the prefix for all agent log files.
- *JobID* is the Oracle BI Scheduler job identifier for the agent.
- *InstanceID* is the Oracle BI Scheduler instance identifier for the agent.
- *xxx* is the file extension:
  - .err for agent error log files.
  - .log for debug log files.

The agent error and debug log files are written as separate files for each agent instance that fails to execute. You can use a text editor to view the files. Entries are generally self-explanatory.

The presence of an error log does not necessarily mean that an agent failed completely. For example, suppose an agent delivers content to multiple e-mail addresses. If some addresses are invalid or the mail server is down, then an error log is generated for the agent.

You can also view error messages and exit codes for job instances in Job Manager (for more information, see Section 29.2.5, "Instance Properties in Job Manager"). Exit status shows the number of deliveries successfully completed.

9 Managing Usage Tracking

The Oracle BI Server supports the collection of usage tracking data. When usage tracking is enabled, the Oracle BI Server collects usage tracking data for each query and writes statistics to a usage tracking log file or inserts them directly to a database table. It is strongly recommended that you use direct insertion instead of writing to a log file.

> **Note:** A sample usage tracking implementation is provided with the Oracle Business Intelligence installation at:
>
> *ORACLE_ INSTANCE*\bifoundation\OracleBIServerComponent\coreapplication_obis*n*\sample\usagetracking

If you are upgrading from previous versions of Usage Tracking, then see the usage tracking topics in *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

This chapter contains the following topics:

- Section 9.1, "Setting Up Usage Tracking"
- Section 9.2, "Description of the Usage Tracking Data"

9.1 Setting Up Usage Tracking

The Oracle BI Server supports the accumulation of usage tracking statistics that can be used in a variety of ways such as database optimization, aggregation strategies, or billing users or departments based on the resources that they consume. The BI Server tracks usage at the detailed query level.

When you enable usage tracking, statistics for every query are inserted into a database table or are written to a usage tracking log file. If you use direct insertion, then the BI Server directly inserts the usage tracking data into a relational database table. It is recommended that you use direct insertion to write statistics to a database table.

When the BI Server starts, it validates the column names in the metadata against the list of valid columns in the usage tracking table. The following events occur:

- **Column names.** If there is a mismatch between the columns in the database table and the columns in the metadata, then it results in a database error on insert.
- **Varchar length.** If the length in the metadata and the set length in the table do not match, then an error is written to the nqserver.log file and usage tracking is disabled.

This section contains the following topics:

- Section 9.1.1, "Setting Up Direct Insertion to Collect Information for Usage Tracking"
- Section 9.1.2, "Setting Up a Log File to Collect Information for Usage Tracking"

9.1.1 Setting Up Direct Insertion to Collect Information for Usage Tracking

Direct insertion is the recommended method for setting up usage tracking. To set up direct insertion, update the following parameters in the Usage Tracking section of the NQSConfig.INI file:

- **ENABLE.** Set this parameter to `YES` to enable usage tracking.
- **DIRECT_INSERT.** This parameter determines whether the query statistics are inserted directly into a database table or are written to a file for subsequent loading. Set this parameter to `YES` to enable direct insertion.
- **PHYSICAL_TABLE_NAME.** To insert query statistic information into a table, you must provide the name of the table and the connection pool that is used to access the table (see the CONNECTION_POOL parameter).

  The fully qualified physical table name consists of as many as four components (database name, catalog name, schema name, and table name). Each component is surrounded by double quotes (") and separated by a period (.). The physical table name must be fully qualified. This fully qualified physical table name must match a table name in the physical layer of the loaded repository. The following is an example of a physical table name for the Usage Tracking table in the Oracle Business Intelligence repository:

  ```
  PHYSICAL_TABLE_NAME = "Oracle BI Usage"."Catalog"."dbo"."S_NQ_ACCT" ;
  ```

  In this example, `Oracle BI Usage` represents the database component, `Catalog` represents the catalog component, `dbo` represents the schema component, and `S_NQ_ACCT` represents the table name.
- **CONNECTION_POOL.** The fully specified connection pool name has two parts: database name and connection pool name. Each part is surrounded by double quotes (") and separated by a period (.). The fully qualified connection pool name should match a connection pool name in the physical layer of the loaded repository. For an example, see the following connection pool name in the Oracle Business Intelligence repository:

  ```
  CONNECTION_POOL = "Oracle BI Usage"."Connection Pool" ;
  ```

  In this example, `Oracle BI Usage` represents the database component and `Connection Pool` represents the actual connection pool name.

  For Usage Tracking insertions to succeed, the connection pool must be configured with a user ID that has write access to the back-end database.

  > **Note:** It is recommended that the connectivity type supports international data.

  See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about configuring connection pools.
- **BUFFER_SIZE.** This parameter indicates how much memory the BI Server should allocate for buffering the insert statements. Such a buffer lets the BI Server submit

multiple insert statements as part of a single transaction, improving Usage Tracking insert throughput. It also means that ordinary analyses do not have to wait on Usage Tracking insertions, which improves average query response time. You might want to adjust this value based on available memory and memory utilization on the server computer.

- **BUFFER_TIME_LIMIT_SECONDS.** This parameter indicates the maximum amount of time that an insert statement remains in the buffer before the Usage Tracking subsystem attempts to issue it. This time limit ensures that the BI Server issues the insert statements quickly, even during periods of extended quiescence.
- **NUM_INSERT_THREADS.** This parameter indicates the number of threads that remove insert statements from the buffer and issue them to the Usage Tracking database. Assuming separate connection pools for readers and inserters, the number of insert threads should typically equal the Maximum Connections setting in the connection pool.
- **MAX_INSERTS_PER_TRANSACTION.** This parameter indicates the maximum number of insert statements that the Usage Tracking subsystem attempts to issue as part of a single transaction. The larger this number, the greater potential throughput for UsageMarathon Tracking inserts. However, a larger number also increases the likelihood of transactions failing due to deadlocks. A small value for `BUFFER_TIME_LIMIT_SECONDS` can limit the number of inserts per transaction.

See Appendix A, "NQSConfig.INI File Configuration Settings" for additional information about the usage tracking configuration parameters.

9.1.2 Setting Up a Log File to Collect Information for Usage Tracking

Log file collection is an alternate method for configuring usage tracking. Although this feature is supported, it is recommended that you use direct insertion to collect information for usage tracking. See Section 9.1.1, "Setting Up Direct Insertion to Collect Information for Usage Tracking" for details.

The following sections explain how to configure log file collection for usage tracking. Before you configure the log file parameters, you must set the `ENABLE` parameter in the Usage Tracking section of the NQSConfig.INI file to `YES` to enable usage tracking. See Appendix A, "NQSConfig.INI File Configuration Settings" for more information about the usage tracking configuration parameters.

This section contains the following topics:

- Section 9.1.2.1, "Selecting an Output Location"
- Section 9.1.2.2, "File Naming Conventions"
- Section 9.1.2.3, "Output File Format"
- Section 9.1.2.4, "Performance Considerations"

9.1.2.1 Selecting an Output Location

The `STORAGE_DIRECTORY` parameter in the Usage Tracking section of the NQSConfig.INI file determines the location of usage tracking log files. If usage tracking is enabled, but no storage folder is specified, then the files are written in the logs folder in the software installation folder (for example, in \OBI11g\logs).

Current files are periodically written to disk, and new files are created. The `CHECKPOINT_INTERVAL_MINUTES` parameter controls the frequency with which usage tracking data is flushed to disk, and the `FILE_ROLLOVER_INTERVAL_`

MINUTES parameter controls the frequency with which the current usage tracking log file is closed and a new file is created.

When usage tracking is enabled, every query is logged to a usage tracking log file. This might require a large amount of available storage. For example, assume an average of 300 bytes of data output for each query and 10 queries per second over an 8 hour day. This results in approximately 83 MB of usage tracking data that is written to storage per day. If this example is extended to a 24 x 7 operation, then the result is approximately .25 GB of storage per day.

The BI Server has no limit on the size or quantity of usage tracking log files that can exist in the specified location. You must ensure that sufficient space is available and that you remove or archive old usage tracking files.

Caution: Insufficient storage space might cause you to lose usage tracking data. If the BI Server encounters an error while accessing a usage tracking output file, then it immediately discontinues the collection of usage tracking statistics and issues an error message to the nqserver.log and, on Windows, to the Windows Event log. Even if additional storage space is made available, the collection of usage tracking statistics does not resume until the server is restarted.

9.1.2.2 File Naming Conventions

The file naming scheme for the usage tracking log files is NQAcct.yyyymmdd.hhmmss.log, where yyyy is the year, mm is the month, dd is the day, hh is the hour, mm is the minute, and ss is the second of the timestamp when the file was created. For example, if the server creates the usage tracking log file at 07:15:00 AM on February 12, 2010, the filename would be NQAcct.20100212.071500.log. After the specified rollover interval, this file is flushed to disk and closed and a new log file, with the current date and timestamp, is created.

9.1.2.3 Output File Format

The usage tracking log files are text files, in semicolon-delimited (;) format. A semicolon is used as the column delimiter because the logical SQL text contains commas. A line feed delimits the end of each row of data.

While there is no guaranteed unique key for the usage tracking data, a combination of User name, Node ID, Start timestamp, and Query text is usually sufficient. Query_Key can be used as an unique key but Query_Key is available only in direct insert.

For information about sample scripts to help you extract data from usage tracking log files and load it to appropriately formatted relational database tables, see Chapter 9, "Managing Usage Tracking." See also Section 9.2, "Description of the Usage Tracking Data" for more information about the contents of each column.

Table 9–1 shows the format of the usage tracking output file.

Table 9–1 Usage Tracking Output File Format

Column Number	Column Name	Data Type	Max Data Size	Nullable
1	User name	Varchar	128	No
2	Repository name	Varchar	128	No
3	Subject area name	Varchar	128	No
4	Node ID	Varchar	15	No

Table 9–1 (Cont.) Usage Tracking Output File Format

Column Number	Column Name	Data Type	Max Data Size	Nullable
5	Start timestamp	Char (Timestamp)	19	No
6	Start date	Char (yyyy-mm-dd)	10	No
7	Start hourMin	Char (hh:mm)	5	No
8	End timestamp	Char (Timestamp)	19	No
9	End date	Char (yyyy-mm-dd)	10	No
10	End hourMin	Char (hh:mm)	5	No
11	Query Text	Varchar	1024	No
12	Success indicator	Integer	4	No
13	Row count	Integer	4	Yes
14	Total time (secs)	Integer	4	Yes
15	Compilation time (secs)	Integer	4	Yes
16	Number of database queries	Integer	4	Yes
17	Cumulative db time (secs)	Integer	4	Yes
18	Cumulative db rows	Integer	4	Yes
19	Cache indicator	Char	1	No
20	Query source	Varchar	30	No
21	Presentation Catalog path	Varchar	250	No
22	Dashboard name	Varchar	150	Yes

Table 9–1 describes the schema. The following list contains more detail about Integer data types, Timestamp columns, and the Query Text column:

- **Integers data types.** All data in the output file is in character format. The data in columns 12 through 18 are output as text representations of integer numbers. Therefore, they behave more like Varchar(10) columns than integers. For example, if the row count is one million rows, then 1000000 appears in the output file in column 13 (Row count). This constitutes seven bytes of data, even though the data represents a 4-byte internal integer value.

  In column 12, a Success indicator value of 0 signifies a successful query. All nonzero values indicate failure. The following failure indicators are currently defined:

  – 1 indicates timeout

  – 2 indicates row limit violation

  – 3 indicates unknown error

  The subsequent integer columns are valid only if the Success indicator (column 12) signifies a successful query (value is 0):

- **Timestamp columns.** The Start timestamp and End timestamp columns indicate the wall clock time (local time) when the logical query started and finished. Each value is 19 bytes of character data that represents a SQL-92 timestamp. The format is yyyy-mm-dd-hh:mm:ss. The related columns, Start date and End date, contain just the date component from the respective timestamps (in the yyyy-mm-dd format). Finally, the related columns, Start hourMin and End hourMin, contain just the hour and minute components from the respective timestamps (in a char hh:mm format).

9.1.2.4 Performance Considerations

When usage tracking is enabled, the BI Server collects usage tracking data for every query. This data, however, is written to disk only at user-specified intervals, known as checkpoints. The default setting is to have checkpoints every 5 minutes.

While you can modify this value in the NQSConfig.INI file, reducing the interval adds overhead and, if set low enough, could potentially impact server performance. Setting the value higher increases the amount of usage tracking data that could be lost in the unlikely event of an abnormal shutdown of the BI Server.

The BI Server periodically initiates usage tracking log file rollovers. A rollover consists of closing the current usage tracking log file and opening a newly created one for writing subsequent data. The frequency at which rollovers occur is called a rollover interval. The default rollover interval is 240 minutes (every 4 hours).

Usage tracking log files that are closed are available for analysis. Setting a lower rollover interval makes usage tracking log files available for analysis sooner, but at the cost of additional overhead.

If the checkpoint interval equals or exceeds the rollover interval, then only the rollover occurs explicitly. The checkpoint occurs implicitly only when the old usage tracking log file is closed.

9.2 Description of the Usage Tracking Data

Table 9–2 describes each column in the usage tracking table. Where appropriate, the data type and length is also included.

Table 9–2 Usage Tracking Data

Column	Description
CACHE_IND_FLG	Default is N. Y indicates a cache hit for the query; N indicates a cache miss.
COMPILE_TIME_SEC	The time in seconds that is required to compile the query.
CUM_DB_TIME_SEC	The total amount of time in seconds that the BI Server waited for back-end physical databases on behalf of a logical query.
CUM_NUM_DB_ROW	The total number of rows that are returned by the back-end databases.
END_DT	The date the logical query was completed.
END_HOUR_MIN	The hour and minute the logical query was completed.
END_TS	The date and time the logical query finished. The start and end timestamps also reflect any time that the query spent waiting for resources to become available. **Note:** If the user submitting the query navigates away from the page before the query finishes, then the final fetch never happens, and a timeout value of 3600 is recorded. However, if the user navigates back to the page before the timeout, then the fetch completes at that time, and this is recorded as the end_ts time.

Table 9–2 (Cont.) Usage Tracking Data

Column	Description
ERROR_TEXT	Default is Null. Varchar(250) Error message from the back-end database. This column is only applicable if the `SUCCESS_FLG` (for more information, see entry later in this table) is set to a value other than 0 (zero). Multiple messages are concatenated and are not parsed by the BI Server.
NODE_ID	The host name of the computer where the BI Server is running.
NUM_CACHE_HITS	Default is Null. Number(10,0). For DB2, the data type and length is Decimal(10,0). Indicates the number of times that the cache result returned for query.
NUM_CACHE_ INSERTED	Default is Null. Number(10,0). For DB2, the data type and length is Decimal(10,0). Indicates the number of times that the query generated a cache entry.
NUM_DB_QUERY	The number of queries that were submitted to back-end databases to satisfy the logical query request. For successful queries (SuccessFlag = 0) this number is 1 or greater.
PRESENTATION_NAME	Default is Null. Varchar(128) The name of the Oracle BI Presentation Catalog.
QUERY_BLOB	The data type is `ntext` when using SQLServer, and is `CLOB` when using ORACLE, DB2, or TERRADATA databases. Contains the entire logical SQL statement without any truncation.
QUERY_KEY	Default is Null. Varchar(128). An MD5 hash key that is generated by Oracle Business Intelligence from the logical SQL statement.
QUERY_SRC_CD	The source of the request (for example, Drill or Report).
QUERY_TEXT	Varchar (1024). The SQL statement that was submitted for the query. You can change the length of this column (using the ALTER TABLE command), but note that the text that is written into this column is always truncated to the size that is defined in the physical layer. It is the responsibility of the repository administrator not to set the length of this column to a value greater than the maximum query length that is supported by the back-end physical database. For example, Oracle databases enable a maximum Varchar of 4000, but Oracle truncates to 4000 bytes, not 4000 characters. Hence, if you use a multibyte character set, the actual maximum string size has a varying number of characters, depending on the character set and characters used.
REPOSITORY_NAME	The name of the repository that the query accesses.
ROW_COUNT	The number of rows that are returned to the query client. **Note:** When a large amount of data is returned from a query, this column is not populated until the user displays all of the data.
IMPERSONATOR_USER_ NAME	Default is Null. Varchar(128) The user name of the impersonated user. If the request is not run as an impersonated user, then the value is NULL.

Table 9–2 (Cont.) Usage Tracking Data

Column	Description
SAW_DASHBOARD	The path name of the dashboard. If the query was not submitted through a dashboard, then the value is NULL.
SAW_DASHBOARD_PG	Default is Null. Varchar(150) The page name in the dashboard. If the request is not a dashboard request, then the value is NULL.
SAW_SRC_PATH	The path name in the Oracle BI Presentation Catalog for the analysis.
START_DT	The date that the logical query was submitted.
START_HOUR_MIN	The hour and minute that the logical query was submitted.
START_TS	The date and time that the logical query was submitted.
SUBJECT_AREA_NAME	The name of the business model that is being accessed.
SUCCESS_FLG	The completion status of the query, as defined in the following list: ▪ 0 - The query completed successfully with no errors. ▪ 1 - The query timed out. ▪ 2 = The query failed because row limits were exceeded. ▪ 3 = The query failed due to some other reason.
TOTAL_TIME_SEC	The time in seconds that the BI Server spent working on the query while the client waited for responses to its analyses. This setting is the same as the Response time in the nqquery.log file, as described in Section 8.3.1.1, "Setting the Query Logging Level."
USER_NAME	The name of the user who submitted the query.

Part V

Configuring Oracle Business Intelligence

Although the installer installs Oracle Business Intelligence with a functional sample application, some functionality requires additional configuration changes (for example, the specification of connection details to external systems and e-mail systems). You can also modify default configuration settings to adapt Oracle Business Intelligence to your environment and user needs.

This part contains the following chapters:

10

Configuring Repositories

You can use Fusion Middleware Control to publish the Oracle BI metadata repository and set the location of the Oracle BI Presentation Catalog. Publishing a repository enables the Oracle BI Server to load the repository into memory upon startup and makes the repository available for queries.

When you upload a repository file for publishing, you provide the name and location of the repository to upload and the current repository password.

The published repository is available to all Oracle BI Server clients, not just to Oracle BI Presentation Services.

This chapter contains the following topics:

- Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location"
- Section 10.2, "Understanding Repository Version Numbers"
- Section 10.3, "Using the System MBeans to Publish the Metadata Repository"

10.1 Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location

In addition to publishing the metadata repository, you can use Fusion Middleware Control to provide a shared location for the repository and a local or shared location for the Oracle BI Presentation Catalog.

You can publish the repository in *distributed mode* or *shared mode*:

- In distributed mode, each Oracle BI Server instance loads the repository from its default local directory. Repositories uploaded through Fusion Middleware Control are uploaded to each default local directory. Any changes made to the repository using the Administration Tool affect only the Oracle BI Server component that is associated with the changed repository file.
- In shared mode, each Oracle BI Server instance loads the repository from the shared network location designated as the repository publishing directory. Note that repositories uploaded through Fusion Middleware Control are uploaded to both the shared location and the default local directory for each Oracle BI Server. You typically publish the repository in shared mode when you have multiple Oracle BI Server components in your deployment.

Like the Oracle BI repository, the Oracle BI Presentation Catalog can be published in shared mode, or distributed mode. Each Presentation Services instance loads the Oracle BI Presentation Catalog from the catalog location specified in Fusion Middleware Control.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to upload a repository and set the location of the Oracle BI Presentation Catalog:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Repository tab of the Deployment page.

   On the Repository tab, you can view the name of the current published repository (called the **Default RPD**). Note that the Data Source Name (DSN) visible to Oracle BI Server clients for the repository published through Fusion Middleware Control is always **Star**.
3. Click **Lock and Edit Configuration** to allow changes to be made.
4. To publish the repository in shared mode, select **Share Repository** and provide a **Shared Location**. If you do not select this option, then the repository that you upload on this page is published in distributed mode by default.
5. To upload a repository and to set the location for the Oracle BI Presentation Catalog, complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
   - **Repository File** option
   - **Repository Password** option
   - **Catalog Location** option
6. Click **Apply**, then click **Activate Changes**.
7. Return to the Business Intelligence Overview page and click **Restart**.

See Appendix A, "NQSConfig.INI File Configuration Settings" for information about advanced configuring settings for repositories. Note that hosting multiple repositories on a single Oracle BI Server is not recommended for production systems.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

10.2 Understanding Repository Version Numbers

When you upload a repository, it is copied to the local repository directory for each Oracle BI Server component, located at:

ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_obis*n*\repository

If you have published the repository in shared mode, then the repository is also copied to the shared network location.

After the repository is uploaded, it is given a version number suffix (for example, SampleApp_bi001.rpd). This version number is incremented each time that the repository is uploaded. To revert to a previous version, you must re-upload the specific version of the repository to use.

Note that you can delete older versions from the repository directory or shared network directory to reclaim disk space.

10.3 Using the System MBeans to Publish the Metadata Repository

This section explains how to use the Fusion Middleware Control System MBean Browser to publish the default metadata repository.

To use the System MBean Browser to publish the Oracle BI repository:

1. Manually copy the repository file to publish to the local repository directory for each Oracle BI Server component in the deployment, at:

   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_obis*n*\repository

2. Use the System MBean Browser to specify the new repository name. See Section 3.5, "Using the Fusion Middleware Control MBean Browser to Update Configuration Settings" for more information.

3. Commit your changes and restart Oracle Business Intelligence.

Note that if you perform this procedure and then later scale out the deployment to include additional Oracle BI Server components, then you must manually copy the repository file to the local repository directory for the new Oracle BI Server component. Perform this copy step *after* the Oracle BI Server component has been created, but *before* restarting the system.

11

Configuring Connections to External Systems

This chapter explains how to configure connections to systems that are external to Oracle Business Intelligence. This chapter contains the following topics:

- Section 11.1, "Configuring E-Mail and Agents"
- Section 11.2, "Configuring for Actions with the Action Framework"
- Section 11.3, "Configuring for Connections to the Marketing Content Server"
- Section 11.4, "Configuring to Download Oracle Hyperion Smart View for Office"

As part of the process of configuring connections to external systems, you can configure a database for the Oracle BI Scheduler. See Section 25.2, "Configuring a Database for the Oracle BI Scheduler" for information.

11.1 Configuring E-Mail and Agents

You can use Fusion Middleware Control to configure common e-mail settings that are used by agents. Advanced configuration settings are described in Chapter 20, "Configuring and Managing Agents."

11.1.1 Using Fusion Middleware Control to Configure Oracle BI Scheduler E-mail Settings That Affect Agents

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to configure Oracle BI Scheduler e-mail settings that affect agents:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Mail tab of the Deployment page as appropriate.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
   - **SMTP Server**

- **Port**
- **Display name of sender**
- **Email address of sender**
- **Username**
- **Password**
- **Confirm password**
- **Number of retries upon failure**
- **Maximum recipients**
- **Addressing method** To, Blind Copy Recipient (Bcc)
- **Use SSL**
- **Specify CA certificate store**
- **CA certificate file**
- **SSL certificate verification depth**
- **SSL cipher list**

5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

See Chapter 20, "Configuring and Managing Agents" for information about advanced configuring settings for agents.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

11.2 Configuring for Actions with the Action Framework

Users can create *actions* in the Oracle BI Presentation Services user interface. An action is an operation or process that can be invoked explicitly by a user clicking an action link. Actions can also be invoked automatically, as the final step of an agent.

You can configure for the use of actions in your organization. For a comprehensive discussion of how to use the Action Framework to enable actions for external systems, including a complete description of each configuration setting and detailed examples, see *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition*.

11.3 Configuring for Connections to the Marketing Content Server

Oracle Marketing Segmentation handles segmentation, which involves dividing a target audience into different segments given different criteria based on the subject areas. When a segment is ready, users create lists of the contacts and accounts that satisfy the criteria for the segment. Users then specify whether to store the generated lists on the file system, in a database, or on a specified Content Server.

For users to store the lists on a Content Server, you as the administrator must configure the connection to the Content Server by specifying the appropriate URL and credentials. You use Fusion Middleware Control to make this specification. You can also manually specify additional settings in the instanceconfig.xml file.

You can configure for connections to the Marketing Content Server, as described in the following sections:

- Section 11.3.1, "Using Fusion Middleware Control to Configure the Connection to the Marketing Content Server"
- Section 11.3.2, "Manually Editing Settings for the Marketing Content Server Connection"

11.3.1 Using Fusion Middleware Control to Configure the Connection to the Marketing Content Server

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to configure the connection to the Marketing Content Server:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Marketing tab of the Deployment page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
   - **URL** option
   - **Username** option
   - **Password** option
   - **Confirm Password** option
5. Click **Apply**, then click **Activate Changes** to execute your changes and release the lock to enable another system administrator to make changes
6. Return to the Business Intelligence Overview page and click **Restart**.

For information on using methods in the Oracle BI Systems Management API to configure the connection to the content server, see Chapter 30, "Introducing the Oracle BI Systems Management API."

11.3.2 Manually Editing Settings for the Marketing Content Server Connection

Use various elements in the instanceconfig.xml file to change additional settings for the connection to the Marketing Content Server.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit additional settings for the Marketing Content Server connection:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the elements that are described in the following list:

- SocketTimeoutSec: Specifies the number of seconds that the socket waits for the Content Server to respond while transferring records. The default value is 60. There is no minimum or maximum value.
- FileSizeMB: Specifies the size in megabytes of files that are generated during ListGeneration and inserted into the Content Server. The default is 10. The minimum size is 1MB and the maximum size is 50MB.

3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <MarketingContentServer>
    <URL>myhost.com:6666/st1b2rep/idcplg</URL>
    <SocketTimeoutSec>120</SocketTimeoutSec>
    <FileSizeMB>5</FileSizeMB>
  </MarketingContentServer>
</ServerInstance>
```

   You do not specify values for user name and password in the instanceconfig.xml file. Instead you specify values in Fusion Middleware Control that are stored securely within the central credentials wallet, along with all other user names and passwords.

   Also, do not change the URL element. This value is centrally managed by Fusion Middleware Control.

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

11.4 Configuring to Download Oracle Hyperion Smart View for Office

The Oracle BI EE Home page provides a link to download and install Oracle Hyperion Smart View for Office. Complete the following steps in the specified order to configure for users to download Smart View, which was installed as part of EPM System 11. While the order of installation is not important, Provider Services must be started before users can connect to Oracle BI EE using Smart View.

To configure to download Smart View:

1. Install and start Provider Services. Smart View uses Provider Services for accessing Oracle BI EE content.

   For information on installing Provider Services, see the *Oracle Hyperion EPM System Installation Start Here Guide*.

2. Install Oracle BI EE.

   For information, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

3. Configure a connection to Oracle BI EE using the Smart View Data Source Manager in the Smart View client.

   For information, see the chapter on connecting to data sources in the *Oracle Smart View for Office, Fusion Edition*.

Note: There is no separate administration tool for configuring data sources for use with Smart View. Administrators can install Smart View using the standalone installer, if they want to set up connections before enabling the Smart View download for users. The assumption is that the Oracle BI EE installation has been completed and the services run along with the Provider Services.

4. Open the instanceconfig.xml file for editing, as described in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings."
5. Update the SmartViewInstallerURL element with the URL of the location from which users can download the Smart View Client installation file, as shown in the following example:

```
<WebConfig>
  <ServerInstance>
    <SmartViewInstallerURL>http://<URL to download SmartView
client>/</SmartViewInstallerURL>
  </ServerInstance>
</WebConfig>
```

6. Ensure that the appropriate users have the Access to Oracle BI for Microsoft Office privilege specified on the Privilege Administration page. This privilege controls access to downloads for both Oracle Business Intelligence Add-in for Microsoft Office and Smart View.

   For more information, see Section C.2.3.3.2, "Access to Oracle BI for Microsoft Office Privilege."
7. Restart Oracle BI Presentation Services to see the download option for Smart View.

12 Configuring Presentation Setting Defaults

You use Fusion Middleware Control to change default presentation settings that administrators commonly change. Advanced configuration settings are described in Section 19.4, "Manually Changing Presentation Settings."

12.1 Using Fusion Middleware Control to Change Presentation Setting Defaults

Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to change presentation setting defaults:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Presentation tab of the Deployment page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
   - **Show page tabs** option
   - **Show section headings** option
   - **Allow dashboard sections to be collapsible** option
   - **Pivot Tables show auto-preview** option
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

See Chapter 19, "Configuring and Managing Analyses and Dashboards" for information about advanced configuring settings for analyses and dashboards.

For information on using methods in the Oracle BI Systems Management API to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

13

Configuring Mapping and Spatial Information

When you install Oracle BI Enterprise Edition, you install functionality for users to see maps that display data. Before users can see maps in analyses and dashboards, you must understand the system requirements, specify layers and maps, and configure the metadata.

This chapter contains the following sections:

- Section 13.1, "What are the System Requirements for Map Views?"
- Section 13.2, "Administering Maps"

See Chapter 21, "Configuring Advanced Options for Mapping and Spatial Information" for information about advanced configuration options for maps.

13.1 What are the System Requirements for Map Views?

To include map views on dashboards, you must have the following:

- Spatial boundary data. NAVTEQ provides this data to Oracle customers, which can be downloaded from the Oracle Technology Network. This spatial data and any other spatial metadata, including themes and styles, must be stored in an Oracle Database to be accessed by Oracle MapViewer for display in map views.
- Hosted maps. Oracle BI EE allows users to access hosted maps based on NAVTEQ data from the Oracle eLocation service. Terms and conditions of use are located at the following URL:

  `http://elocation.oracle.com/elocation/legal.html`
- Oracle MapViewer, which is a J2EE service for rendering maps using spatial data managed by Oracle Spatial. MapViewer is installed as part of Oracle BI Enterprise Edition and deployed in the same domain as Oracle BI EE on the Web application server. MapViewer provides services and tools that hide the complexity of spatial data queries and cartographic rendering, while providing customizable options for more advanced users. MapViewer is designed to integrate with Location-Based services and applications.

  For information, see Section 21.1, "Configuring MapViewer to Support Map Views."
- Oracle Database, version 10*g* or higher, to store the spatial data.
- Oracle Locator, which is a feature of Oracle Database (all editions) that provides core location functionality needed by most customer applications.

Oracle Spatial is an option for Oracle Database Enterprise Edition that provides advanced spatial features to support high-end geographic information systems (GIS) and location-based services (LBS) solutions. The Spatial option is required only if you plan to make use of maps or features that require advanced spatial capabilities that are not available with the Locator option. Additional Oracle Spatial features include raster and 3D data management, spatial Web services, topology, network data modeling, resource description framework (RDF), and semantic Web capabilities.

Oracle Spatial and Oracle Locator support relevant Open Geospatial Consortium (OGC) standard specifications.

- Metadata of the mapping between Oracle BI EE data and spatial data, which can be stored in a file system, in the Oracle BI Presentation Catalog.

When these pieces are in place, you administer the map using the Oracle BI Presentation Services Administration page, as described in Section 13.2, "Administering Maps."

13.1.1 Hardware Sizing and Deployment Strategy for Maps

Rendering map views is computationally more intensive than rendering tabular views, due to the requirements to:

- Query spatial data.
- Create the polygons and shapes that correspond to geographical entities such as countries and states.
- Place the polygons and shapes on a background map.
- Provide end-user interactivity such as the ability to pan and zoom, to adjust color thresholds, and to show or hide formats.

You should assess the extent of expected usage of map views at your organization including the number of users that are expected to use map views, the amount of data to be displayed on map views, and the amount of spatial data that is expected to be displayed (such as only city boundaries or street level details). Based on this assessment, decide on an appropriate hardware sizing and deployment strategy. You should also review the available documentation on best practices for achieving optimal performance and scalability with your Oracle MapViewer deployment.

13.2 Administering Maps

Before content designers can create map views, you as the Oracle BI Administrator must specify layers and maps and configure the metadata. This section discusses the following:

- Section 13.2.1, "Working with Maps and Layers"
- Section 13.2.2, "Administration Page Functions"
- Section 13.2.3, "Administering Maps"

13.2.1 Working with Maps and Layers

The first step is to select the layers for use on maps. An administrator has configured layers using the Map Builder tool of Oracle Spatial. You next select at least one map from a list of those that an administrator has configured using the Map Builder tool of Oracle Spatial. This map becomes the background on which the layers are applied.

You can optionally specify images for use with map formats. This section provides the following information about maps and layers:

- Section 13.2.1.1, "Associating Layers with Columns"
- Section 13.2.1.2, "Ordering Layers on Maps"

13.2.1.1 Associating Layers with Columns

After selecting layers and maps, you associate those layers with columns in the subject area folders. If the association between a column and a layer is incorrect, then the layer cannot be displayed correctly on the map. The association ensures that shape definitions can be found in the database for the column values during map rendering. You must ensure that a shape geometry definition exists for each column value in the database. If a shape geometry definition does not exist for a particular column value, then the shape cannot be shown on the map and might inhibit user interactions on the map.

Shape lookup is based on the column values, and column-to-layer mapping is independent of locale or language. Therefore, you must ensure that the spatial column that is being associated with a layer does not itself have values that are affected by locale or language. To ensure this association, do one of the following:

- Model spatial columns as double columns in the business modeling layer, which is the recommended method.
- Create special spatial columns that have values that do not change across locale or language. You must ensure that content designers are not confused seeing these special columns in the Subject Areas pane when working with analyses.

The advantages of using double columns are the following:

- You can provide code values (that is, descriptor IDs) for each shape definition and at the same time show display values (that is, descriptor values) according to locale or language.
- Code values are passed only as layer key values for lookup.
- You eliminate the need for the complex joining of various columns to uniquely locate the shape. For example, a layer geometry table might contain multiple cities with the name of London. To uniquely distinguish these cities, you might use a pattern of Country_State_City, as in US_Kansas_London or Canada_Ontario_London. The problem with this approach is that it might require three separate columns to be grouped, joined by a delimiter (such as an underscore), and associated with a layer. This association requires the content designer to select three columns (Country, State, City) in the criteria to create a single layer.

A layer can be associated with multiple columns in multiple subject areas. You must ensure that a layer is associated with at least one spatial column. If the layer association is missing, then the map might not be updated when a user drills on the mapped BI column.

13.2.1.2 Ordering Layers on Maps

The ordering of map layers is very important. You must pay close attention to ensure that users have a seamless experience while navigating on the map (that is, drilling and zooming). In the Edit Background Map dialog, you assign each layer a minimum and maximum zoom range. Given that the map zoom slider can slide only from bottom to top vertically, the layers with lower minimum zoom levels are placed at the bottom of the slider. Ensure that the layer grid on the Interactive BI Layers section of the dialog follows a similar pattern, so that you place layers with lower minimum zoom levels at the bottom of the list. Ensure that you sort the layers (by clicking on the sort icon) before closing the dialog.

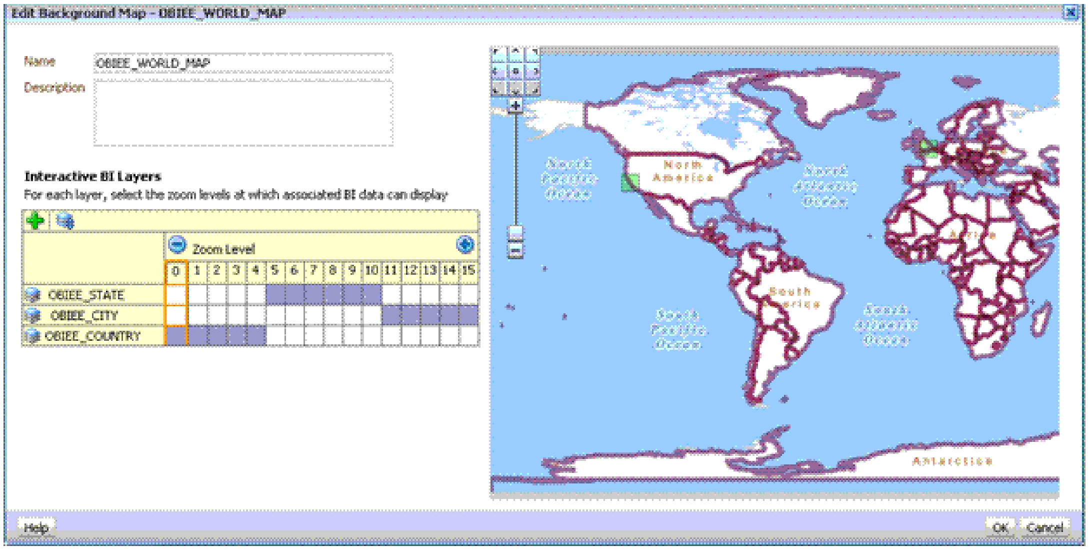
Layer ordering becomes irrelevant when the zoom ranges of layers do not intersect on the scale. Ordering becomes very important when layers have a common minimum and maximum zoom range. Use care to ensure that detailed layers are not hidden by the aggregated layers during drilling or zooming operations.

Example 13–1 World Map with Three Layers

Suppose that a world map has three layers (Country, State, and City) and 15 zoom levels defined on it. The Country layer has a minimum and maximum zoom range of 0-5, the State layer range is 6-10, and the City layer range is 11-15. As the user navigates from the minimum to the maximum zoom level on the map, the layer order (also known as the visual order) is displayed as Country, State, and City.

Figure 13–1 shows the Edit Background Map dialog with the Interactive BI Layers section specified for this example. Reading from bottom to top, you see the layer order in that section as Country, City, and State. Even though the layer order configuration and their respective zoom levels are not consistent with each other, the end user's perception is not affected.

Figure 13–1 Edit Background Map Dialog with No Intersection



Edit Background Map - OBIEE_WORLD_MAP

Name: OBIEE_WORLD_MAP

Description:

Interactive BI Layers

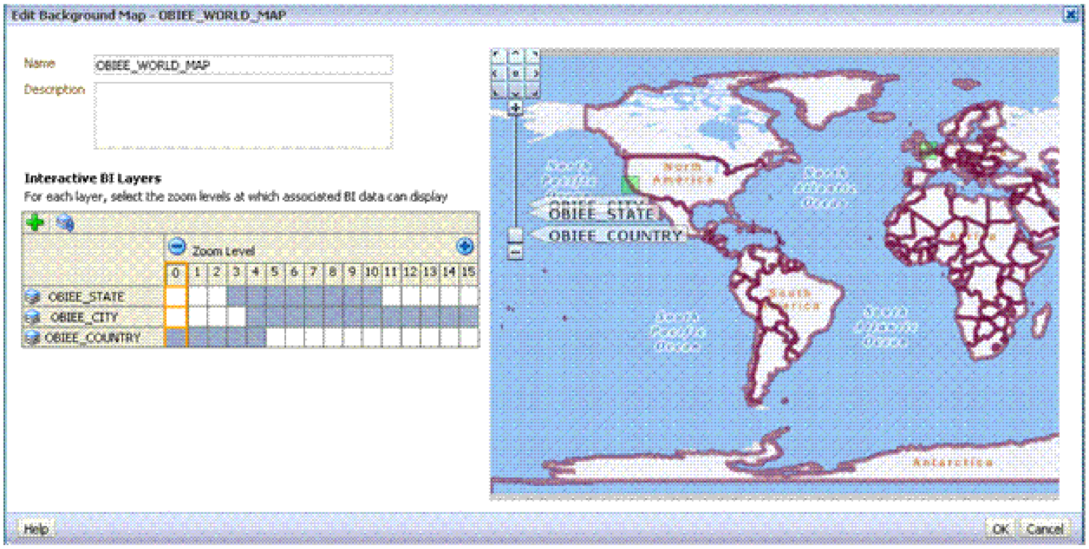
For each layer, select the zoom levels at which associated BI data can display

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OBIEE_STATE						■	■	■	■	■	■					
OBIEE_CITY												■	■	■	■	■
OBIEE_COUNTRY	■	■	■	■	■											

Example 13–2 World Map with Common Levels

Consider the same world map with layers that have common zoom ranges. Suppose that the user zooms to level 4 on the map that corresponds to the Edit Background Map dialog that is shown in Figure 13–2. Notice that all three layers include zoom level 4.

Figure 13–2 Edit Background Map Dialog with Intersection



Edit Background Map - OBIEE_WORLD_MAP

Name: OBIEE_WORLD_MAP

Description:

Interactive BI Layers

For each layer, select the zoom levels at which associated BI data can display

	Zoom Level															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OBIEE_STATE				▨	▨	▨	▨	▨	▨	▨	▨					
OBIEE_CITY					▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
OBIEE_COUNTRY	▨	▨	▨	▨	▨											

The layer ordering at zoom level 4 is unclear, because multiple layers can be shown. Because of the layer order that is specified in the Interactive BI Layers section of the dialog, the map renders with the following order at zoom level 4: Country, City, and State (reading from bottom to top), which is incorrect.

To correct this problem, you must ensure that layers are always ordered properly in the dialog. You can re-order layers by clicking the **Move Down** and **Move Up** buttons on the zoom grid of the dialog, or by clicking the **Sort Layers by Zoom Level** button. For example, you can click the **Sort Layers** button in the dialog to specify that the layer ordering at zoom level 4 is Country, State, and City.

See *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for more information on layers.

13.2.2 Administration Page Functions

The Oracle BI Presentation Services Administration page provides the **Manage Map Data** link. This link displays the Manage Map Data page, where you can manage the logical and display versions of the data from various physical data sources. This defines the layers that content designers use when creating map views. The data that is available for managing maps and data is stored in the Oracle Database as part of MapViewer.

Using this page, you provide:

- Logical names to prevent any existing BI column mappings and map analyses from breaking because of changes to the physical data or to the data source.
- Display names so that the geographic data is meaningful to end users.

13.2.3 Administering Maps

To administer maps:

1. Ensure that Oracle MapViewer is configured properly.

   For information, see *Oracle Fusion Middleware User's Guide for Oracle MapViewer*.

2. Sign in to Oracle Business Intelligence.

3. In the global header, click **Administration**.
4. Click the **Manage Map Data** link to display the Manage Map Data page.
5. Click the Layers tab.
6. Click the **Import Layers** button to display the Import Layers dialog.
7. In the dialog, select the connection and the layers that are needed for zooming and drilling. (This tab is not prepopulated by other selections on the Administration pages. You can use any layers or images with any map.)

   Click **OK** when you have finished selecting layers that are appropriate for the subject area with which you are working.
8. Back on the Layers tab, select a layer, then click the **Edit Layer** button to display the Edit Layer dialog in which you associate layers with attribute columns so that you can display BI data in the map view.

   Click **OK** when you have finished editing the layer.

   You use this tab to associate layers with BI data. If you use the City column in multiple subject areas, then you must associate it with a layer for each subject area.
9. Click the Background Maps tab, then click the **Import Background Maps** button to display the Import Background Maps dialog.
10. In the dialog, select the connection and the main maps to use.

    The connection that you select for the main map can be different from the connection for the layers or images.

    Click **OK** when you have finished selecting main maps.
11. Back on the Background Maps tab, select a map, then click the **Edit Background Map** button to display the Edit Background Map dialog in which you name the map and specify the order of layers and their zoom levels.

    Click **OK** when you have finished editing the map.
12. Optionally, click the Images tab, then click the **Import Images** button to display the Import Images dialog. You can import images if you plan to use them as a format on maps.
13. In the dialog, select the connection and the images to use.

    Click **OK** when you have finished selecting images.
14. Click **OK** when you have finished working with the Administration page.

After you have specified background maps, layers, and the associations between BI columns and spatial columns, MapViewer creates a static image for a map using this information and sends that image for rendering in the browser for use by content designers and end users in map views.

14 Configuring Oracle Scorecard and Strategy Management

If your organization licensed Oracle Scorecard and Strategy Management and if you have the appropriate privileges, then you can use this functionality as part of a default installation with no additional configuration. Oracle Scorecard and Strategy Management also provides the capability to add comments (that is, annotations) or to override the status that is associated with specific dimension values for KPIs, Objectives, and Initiatives. KPI Watchlists offer the capability to add comments or to override statuses for KPIs. To enable these features, you must provide access to a database for storing the comment and status override information.

14.1 Enabling Comments and Status Overrides in Oracle Scorecard and Strategy Management

The database that you installed for use with Oracle Business Intelligence contains the Business Intelligence Platform schema, which includes required Oracle Scorecard and Strategy Management schema tables. For more information about installing a database for Oracle Business Intelligence and running the Repository Creation Assistant (RCU) to create the required schemas, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

To configure Oracle Scorecard and Strategy Management for comments and status overrides:

1. In the Administration Tool, open the repository in online mode.

   Online mode is strongly recommended for performing data access security tasks, such as the task described in Step 12 of this procedure.

2. In the Physical layer, right-click and select **New Database**. The Database dialog box is displayed.
3. For **Name**, enter BSC.
4. For **Database**, select the type of database that you have installed for use with Oracle Business Intelligence (typically Oracle 11g).
5. Select the Connection Pool tab and click the **Add** button. The Connection Pool dialog box is displayed.
6. For **Name**, enter BSC.
7. Select the **Call interface** appropriate for the database (for example, OCI 10g/11g for Oracle Database).

8. For **Data source name**, provide the information that is appropriate for the database that you have installed and configured for use with Oracle Business Intelligence. For example, for Oracle Database, enter a connection string similar to the following:

   ```
   (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.1.100)(PORT=1521))(CONNECT_
   DATA=(SERVER=DEDICATED)(SERVICE_NAME=KPIOracle)(SID=KPIOracl)))
   ```

   When connecting to an Oracle Database data source, you can include the entire connect string, or you can use the net service name defined in the tnsnames.ora file. If you choose to enter only the net service name, then you must set up a tnsnames.ora file in the following location within the Oracle Business Intelligence environment, so that the Oracle BI Server can locate the entry:

   ```
   ORACLE_HOME/network/admin
   ```

9. Select **Shared logon** and enter values for **User name** and **Password**. In this step, you provide the user/schema name and password that you created when you used the Repository Creation Utility (RCU) to populate the Business Intelligence Platform schema in the Oracle Business Intelligence database.

   Ensure that the user that you provide has read/write privileges for the ANNOTATIONS and ASSESSMENT_OVERRIDES tables in the Business Intelligence Platform schema.

10. Click **OK** in the Connection Pool dialog box.

11. Click **OK** in the Database dialog box.

12. Use the Security Manager in the Administration Tool to allow the BISystem application role to execute direct database requests by default for the BSC database object. See "Allowing or Disallowing Direct Database Requests" in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for more information.

13. Save and close the repository.

14. Restart the Oracle BI Server.

15

Configuring Time Zones

If you have users in time zones that are different from the zone for Presentation Services, then you can specify the time stamps that those users see in Oracle Business Intelligence. For example, suppose the server is located in the Pacific time zone in the United States. You can specify that users on the east coast of the United States see time stamps that are displayed in Eastern Standard Time.

If you make no time zone settings and if a user does not specify a preferred time zone using the My Account dialog, then that user sees time displayed according to the local time zone for Presentation Services.

For information on how users specify their preferred time zones, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

This chapter contains the following topics on preferred time zones:

- Section 15.1, "Where Time Zones Are Used"
- Section 15.2, "Setting Time Zones"
- Section 15.3, "Precedence Order for Time Zones"
- Section 15.4, "Where Time Zone Specifications Are Stored"
- Section 15.5, "Description of Time Zone Settings"
- Section 15.6, "Example: Configuration File Settings for Specifying the Time Zone"

15.1 Where Time Zones Are Used

Before you begin to set preferred time zones, see Table 15–1 for information on where time zones are used.

Table 15–1 Time Zone Usage

Type	Description
Oracle BI Presentation Services	As the administrator, you can specify the time zone that Presentation Services uses.
Data from the database	The Oracle BI Administrator specifies the time zone for the data that is retrieved from the database. If you make no time zone settings, then users see the time stamp data in the time zone of the original data as set by the Oracle BI Administrator.

Table 15–1 (Cont.) Time Zone Usage

Type	Description
Content that is displayed in Oracle Business Intelligence	Users who create analyses can specify the time zone that is displayed in their analyses and dashboard prompts. This specification overrides those made by you as the administrator and by end users if they have previously used the column in their queries and have set the time zone. If the specified display time zone supports daylight saving time, then the timestamp values that are displayed are automatically adjusted for daylight saving time.
General time stamps that indicate when events happen	End users can specify the time zone for many general stamps including the following ones: ▪ The scheduled time of agents. ▪ The generated time of alerts or analyses. ▪ The time on which objects in the Oracle BI Presentation Catalog are created, modified, and accessed.
Log files	Log files contain time stamps for various activities.

15.2 Setting Time Zones

Use the following procedure to set time zones for users.

To set preferred time zones for users:

1. Determine the time zone that is set for the server on which Presentation Services is running.
2. Use elements in the Presentation Services configuration file (instanceconfig.xml) or session variables. Consult the following for more information:
   – See Table 15–2 for information on the precedence order for time zones.
   – See Table 15–3 for descriptions of the session variables and elements.
   – See Section 15.6, "Example: Configuration File Settings for Specifying the Time Zone."
   – See the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for complete information on session variables.
3. Encourage end users to specify their preferred time zones using the My Account dialog.
4. Encourage users who create analyses to do the following to set the time stamps for their analyses:
   a. Use the Data Format tab of the Column Properties dialog to specify the time zone that is displayed in the columns of their analyses.
   b. Use the Time Zone dialog to set the time zone that is displayed in dashboard prompts.

15.3 Precedence Order for Time Zones

The actual time zone in which various types of content are displayed follows a precedence order that is described in Table 15–2. In the table, the items with lower numbers override those with higher numbers. For example, Item 1 takes precedence over Item 2.

Table 15–2 Precedence Order for Time Zones

Time Zone For	Determined By
Data	1. The setting of the DATA_TZ session variable. 2. The setting of the DefaultDataOffset element in the instanceconfig.xml file. 3. The time zone of the original data as set by the Oracle BI Administrator (because the time zone is unknown for Presentation Services).
Data display	1. The setting that a content designer makes. 2. The setting of the DATA_DISPLAY_TZ session variable. 3. The setting of the DefaultDataDisplay element in the instanceconfig.xml file. 4. "User-Preferred Time Zone"
General time stamps (not including column data and log files)	1. "User-Preferred Time Zone" 2. The time zone for Oracle BI Presentation Services.
Log file information	1. The setting of the Logging element in the instanceconfig.xml file. 2. The time zone for Presentation Services.

15.3.1 User-Preferred Time Zone

The user-preferred time zone is determined by the following:

1. The specification that a user makes in the My Account dialog.
2. The setting of the TIMEZONE session variable.
3. The setting of the DefaultUserPreferred element in the instanceconfig.xml file.

15.4 Where Time Zone Specifications Are Stored

Whenever a time zone specification is displayed in a list or as the value of a session variable or element in the instanceconfig.xml file, that specification originates from the TimeZones.xml file. This file is stored in the *ORACLE_INSTANCE*\bifoundation\OracleBIApplication\coreapplication\timezone directory.

The TimeZones.xml file contains nearly all time zones from around the world. You should not have to add zones to this file, but you are free to edit this file if you care to. You can choose to delete those zones that users in your organization does not use.

15.4.1 Specifying Time Zone Values

Various editors show the ampersand that appears in time zone values in one of two ways: either the ampersand character itself or its escape sequence. Use care when entering a time zone value, as follows:

- When you use the ampersand in the value of a session variable, include the ampersand character (&) in the value, such as "Pacific Time (US & Canada); Tijuana".

- When you use the ampersand in the value of an element in the Oracle BI Presentation Services configuration file (instanceconfig.xml), include the escape sequence for the ampersand in the value, such as "Pacific Time (US &amp; Canada); Tijuana"

15.5 Description of Time Zone Settings

Table 15–3 describes the session variables and the elements in the instanceconfig.xml file with which you set time zones. When you include elements in the instanceconfig.xml file, you specify the time zone that all users see. When you use session variables, you can specify a different time zone for each user. If you use session variables and you specify values for the appropriate elements in the instanceconfig.xml file, then the values of the session variables override the settings in the instanceconfig.xml file.

Table 15–3 Time Zone Settings

Element	Session Variable	Description	Value
DefaultDataOffset	DATA_TZ	The time zone offset of the original data. To allow the time zone to be converted so that users see the appropriate zone, you must set the value of this element or variable. If you do not set this option, then no time zone conversion occurs because the value is "unknown". For example, suppose you want to convert to Eastern Standard Time (EST), which is Greenwich Mean Time (GMT) - 5. You must specify this value to enable the conversion to EST.	An offset that indicates the number of hours away from GMT time. For example: "GMT-05:00" or "-300", which means minus 5 hours.
DefaultDataDisplay	DATA_ DISPLAY_TZ	Specifies the time zone to use for displaying data. If you do not set this option, then the value is the "User-Preferred Time Zone".	One of the time zones that are specified in the TimeZones.xml file. See Section 15.4.1, "Specifying Time Zone Values."
DefaultUserPreferred	TIMEZONE	Specifies the users' default preferred time zone before they select their own in the My Account dialog. If you do not set this option, then the value is the local time zone from Oracle BI Presentation Services.	One of the time zones that are specified in the TimeZones.xml file. See Section 15.4.1, "Specifying Time Zone Values."

Table 15–3 (Cont.) Time Zone Settings

Element	Session Variable	Description	Value
Logging	na	The time zone of the time stamps that appear in log files that are generated by Presentation Services. If you do not set this option, then the value is the local time zone from Presentation Services	One of the time zones that are specified in the TimeZones.xml file. See Section 15.4.1, "Specifying Time Zone Values."
TimeZone	na	The parent element for the elements that modify the preferred time zone. A child of the ServerInstance element.	na

15.6 Example: Configuration File Settings for Specifying the Time Zone

The following shows a sample section of the instanceconfig.xml file in which the TimeZone element has been added.

```
<TimeZone>
     <DefaultDataOffset>0</DefaultDataOffset>
     <Logging>(GMT-08:00) Pacific Time (US &amp; Canada); Tijuana</Logging>
     <DefaultUserPreferred>(GMT-08:00) Pacific Time (US &amp; Canada);
     Tijuana</DefaultUserPreferred>
     <DefaultDataDisplay>(GMT-06:00) Central Time (US &amp;
     Canada)</DefaultDataDisplay>
</TimeZone>
```

See Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings" for information on editing the instanceconfig.xml file.

16

Localizing Oracle Business Intelligence

Users can easily and dynamically change their language and locale preferences. When users select a language, they see many elements in that language. These elements include user interface components, metadata, messages, and Help files.

This chapter provides information on how to configure for deployment in one or more language environments other than English. This chapter contains the following sections:

- Section 16.1, "What is Localization?"
- Section 16.2, "Localizing Oracle BI Presentation Services"
- Section 16.3, "Setting the Current Locale in the Oracle BI Server"
- Section 16.4, "Localizing Metadata Names in the Repository"
- Section 16.5, "Supporting Multilingual Data"

16.1 What is Localization?

In this chapter, localization refers to the process of adapting the Oracle Business Intelligence deployment to a particular language. If your users speak languages other than English, then use the information in this chapter to adapt your deployment to support multiple languages.

For information on supported languages, see Section 1.8, "System Requirements and Certification".

16.1.1 What Components are Translated?

The following list outlines which components of Oracle Business Intelligence are translated into languages other than English:

- Installer
- Web user interface
- Job Manager interface of the Oracle BI Scheduler
- Catalog Manager
- Oracle BI Presentation Services messages:
  - error
  - warning
  - information

- Oracle BI Server functions:
  - DayName
  - MonthName

> **Note:** If a query is issued using the DayName or MonthName function, but the function is not shipped to a back-end database, then the day or month name is returned in the localized language but the column name remains in English (or might be affected by other localization controls). As an example of this situation, if the LOCALE parameter is set for German, the MonthName query returns the string "Mai" but the column header remains "Name of Month."

- Oracle BI Server and Oracle BI Scheduler messages:
  - error
  - warning
  - information
- Log files:
  - nqserver.log for Oracle BI Server
  - nqquery.log for Oracle BI Server
  - If Clustering is enabled, nQCluster.log for Oracle BI Server Cluster
- Metadata:
  - Dashboards and analyses (Oracle BI Presentation Catalog)
  - Presentation table and column names (.rpd file)
- Oracle BI Administration Tool interface
- ODBC setup

The following list outlines which components of Oracle Business Intelligence are not localized:

- ODBC client tools:
  - nqcmd (UNIX)
  - nQCmd.exe (Windows)
  - nQClient.exe (Windows)

Numerous Oracle Fusion Middleware components, such as Oracle WebLogic Server Administration Console and Fusion Middleware Control, are translated. See Oracle Fusion Middleware documentation for information.

16.1.2 How Do I Localize Oracle Business Intelligence Components?

As the administrator, you perform various tasks to localize the components of Oracle Business Intelligence, as described in the following sections:

- Section 16.2, "Localizing Oracle BI Presentation Services"
- Section 16.3, "Setting the Current Locale in the Oracle BI Server"
- Section 16.4, "Localizing Metadata Names in the Repository"

- Section 16.5, "Supporting Multilingual Data"

16.2 Localizing Oracle BI Presentation Services

As the administrator, you perform various tasks to localize Oracle BI Presentation Services, as described in the following sections:

- Section 16.2.1, "Localizing the User Interface for Oracle BI Presentation Services"
- Section 16.2.2, "Localizing Oracle BI Presentation Catalog Captions"
- Section 16.2.3, "Tip for Arabic and Hebrew in Mozilla Firefox Browsers"
- Section 16.2.4, "Handling the Translation of Layers in Maps"

16.2.1 Localizing the User Interface for Oracle BI Presentation Services

Chapter 22 describes how to customize the user interface for Oracle BI Presentation Services. Part of the process of customizing is localizing the interface, if your users speak languages other than English. Users can select a language on the sign-in page for Oracle BI EE, and many elements of the interface are automatically displayed in the appropriate language. After signing in, users can change the language setting on the Preferences tab of the My Account dialog.

The user's setting is stored in the WEBLANGUAGE session variable. For the Oracle BI Presentation Services user interface, WEBLANGUAGE is set when a user selects a language on the sign-in page.

> **Note:** For Oracle BI Applications, WEBLANGUAGE is set to the language of the user's browser when a user logs in for the first time. For example, if a user with a browser language set to French logs in to Oracle BI Applications for the first time, then the value for WEBLANGUAGE is French, and the metadata is translated to French.

As the administrator, you perform various tasks to localize other elements of the user interface for Oracle BI Presentation Services, as described in the following sections:

- Section 16.2.1.1, "Understanding the Directory Structure for Localizing Oracle BI Presentation Services"
- Section 16.2.1.2, "Localizing Messages for Users' Preferred Currency"
- Section 16.2.1.3, "Specifying the Default Language for the Sign-in Page"
- Section 16.2.1.4, "Specifying the Language in the URL"

16.2.1.1 Understanding the Directory Structure for Localizing Oracle BI Presentation Services

Oracle BI EE is installed with many files that control elements in the user interface and messages. These files are installed in the messages and pages subdirectories of the *ORACLE_HOME*\bifoundation\web\MS-DOS directory. To localize these elements and messages, you copy those files to the l_*xx* subdirectories in the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb subdirectories, where *xx* indicates the language extension. After you have copied the files, you can modify their contents as appropriate for the language that corresponds to the subdirectory in which you have copied them.

16.2.1.2 Localizing Messages for Users' Preferred Currency

Section 17.2.1, "Defining User-Preferred Currency Options Using a Static Mapping" provides a procedure for working with users' preferred currencies. Use the following procedure to localize the messages that are associated with a preferred currency.

To localize the messages that are associated with each users' preferred currency:

1. Go to the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx* directory, where *xx* is the language extension for the language in which you are specifying preferred currencies.
2. In the directory, create a subdirectory called custommessages.
3. In the directory, create a file in XML format, with the name of usercurrencymessages.xml.
4. Add entries such as the following one to this file for the language that corresponds to the directory in which you are working. The following example includes two messages: kmsgMyCurrency1 and kmsgMyCurrency2

   ```
   <?xml version="1.0" encoding="utf-8"?>
   <WebMessageTables xmlns:sawm="com.siebel.analytics.web/message/v1">
   <WebMessageTable system="CurrencyDisplay" table="Messages" code="false">
      <WebMessage name="kmsgMyCurrency1"><TEXT>My Currency Text
   1</TEXT></WebMessage>
      <WebMessage name="kmsgMyCurrency2"><TEXT>My Currency Text
   2</TEXT></WebMessage>
   </WebMessageTable>
   </WebMessageTables>
   ```

5. Edit the userpref_currencies.xml file to specify displayMessage="kmsgMyCurrency1" to use this message.
6. Repeat Steps 1 through 5 for each language for which you must localize these messages.
7. Restart the service for Oracle BI Presentation Services.

In Oracle BI EE, the appropriate localized text is displayed to the user. In this example, the text is My Currency Text 1.

For more information, see Section 22.5, "Customizing the User Interface Using XML Message Files."

16.2.1.3 Specifying the Default Language for the Sign-in Page

The default language in which the Presentation Services sign-in page is displayed is obtained from the user's client browser settings. The following procedure explains how to change the language.

> **Note:** The following procedure uses Internet Explorer 7.0 as an example. If you are using a different browser, then make the necessary substitutions.

To change the default language on a user's login screen in Internet Explorer:

1. In Internet Explorer, from the **Tools** menu, select **Internet Options**.

   The Internet Options dialog is displayed.

2. Click **Languages**.

   The Language Preference dialog is displayed.

   Installed languages are displayed in the Language list. The language at the top of the list is used as the default language.

3. If the desired language is not installed on the browser, then add it.

4. Use the **Move Up** and **Move Down** buttons to position the desired language at the top of the list.

5. Restart the browser and sign into Presentation Services.

   The default language should match the language in the browser's Language list.

> **Note:** If a user does not select a different language from the list on the sign-in page, then the setting for the User Interface Language in the user's My Account dialog determines the language in which the user interface is displayed.

16.2.1.4 Specifying the Language in the URL

When users start Oracle BI EE by displaying the sign-in page, they can select the language as part of the sign-in process. They can also select a language on the Preferences tab of the My Account dialog.

If you provide users with a URL with which they can display a dashboard or other page of the application, then you can define a URL parameter as a profile attribute. Doing so dynamically sets the language of the dashboards and analyses to be consistent with the application's language setting.

For operational applications, symbolic URLs embed dashboards and analyses in the integrated environment. For Oracle BI Presentation Services, the URL parameter Lang designates the language that the Web page renders.

The Lang parameter can be included in the symbolic URL that is defined in the operational application to connect to Oracle Business Intelligence. The Lang parameter is defined as a profile attribute, but when the symbolic URL is constructed at run time, the value is set as the profile attribute LanguageCode. Table 16–1 provides examples of the parameter settings in the Symbolic URL parameters applet, including Lang.

For example, the following URL displays the sign-in page in French.

```
http://Server_Name:port_number/analytics/saw.dll?Dashboard&Lang=fr
```

Table 16–1 Example of Settings in the Symbolic URL Parameters Applet

Name	Type	Path Argument Value	Append	Sequence #
Cmd	Constant	Go	Y	1
Path	Constant	/shared/Sales/Pipeline/Overview/Top 10 Deals	Y	2
nQUser	Command	UseLoginId	Y	3
nQPassword	Command	UseLoginPassword	Y	4
PostRequest	Command	PostRequest	Y	5
Lang	Profile Attribute	LanguageCode	Y	6

16.2.2 Localizing Oracle BI Presentation Catalog Captions

The Oracle BI Presentation Catalog stores objects that users create, such as analyses and dashboards. Text strings hold the names and descriptions of these objects. If you must localize text strings for the objects, then you can export the text strings from the catalog so that they can be translated. You then expose the strings when translation is complete.

The export process creates one XML file for every first-level subfolder in the shared folder, in the format *foldername*captions.xml, where *foldername* is the name of the subfolder in the shared folder. Each XML file contains the text strings for all content in the corresponding first-level folder and its subfolders.

When editing XML files, use an editor that is designed for XML files. Ensure that you follow the encoding that is specified at the top of the XML file and that you escape special characters as appropriate. You and the localization team are responsible for resolving any errors in the translated text strings. Keep in mind that the contents of the catalog are updated whenever objects are added, deleted, or modified.

For example, if the shared folder in the Oracle BI Presentation Catalog contains the first-level folders Marketing, Service, and Sales, then the export process creates three XML files:

- marketingcaptions.xml
- salescaptions.xml
- servicecaptions.xml

After the content is translated, you place these folders in their corresponding location in the following directory:

ORACLE_INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*\captions

Their content loads when the service for Oracle BI Presentation Services starts.

> **Note:** The *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*\captions directory exists only if Oracle Business Intelligence Applications have been installed. If it does not exist, then you must create it.

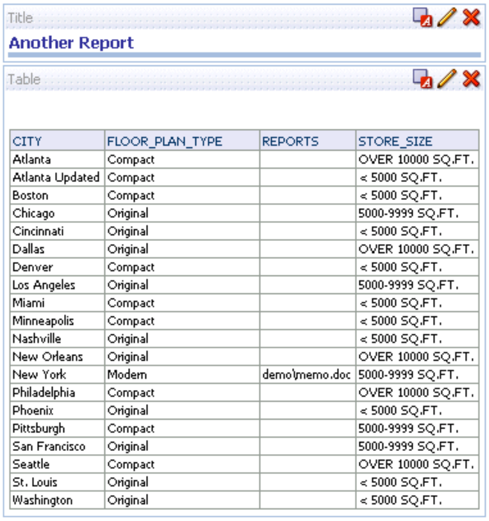
To export catalog text strings:

1. Back up the catalog before exporting from it.

   Ensure that you run the export utility against the actual catalog, not against a copy of the catalog, because the export utility changes the properties of the objects in the catalog against which it runs.

2. In Catalog Manager, open the catalog in either online or offline mode.
3. Select the folder that contains the strings to export. The utility runs against the files in that folder and all its subfolders.

   For example, the title (Another Report) in the following analysis can be exported for translation.



Another Report

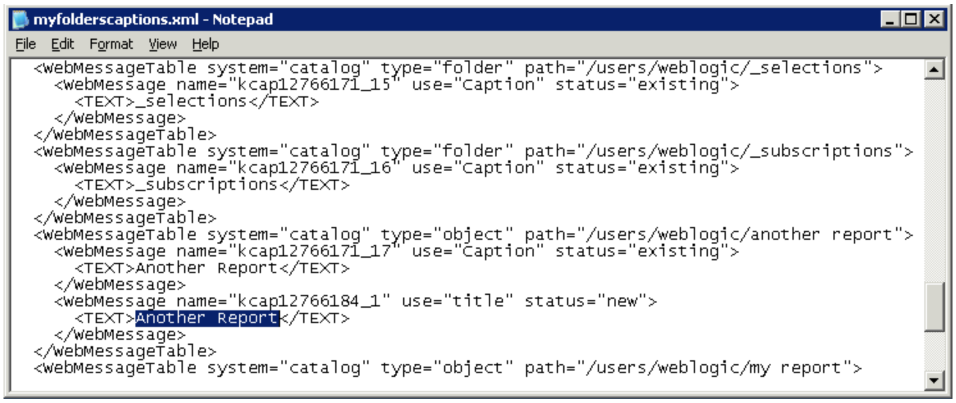
CITY	FLOOR_PLAN_TYPE	REPORTS	STORE_SIZE
Atlanta	Compact		OVER 10000 SQ.FT.
Atlanta Updated	Compact		< 5000 SQ.FT.
Boston	Compact		< 5000 SQ.FT.
Chicago	Original		5000-9999 SQ.FT.
Cincinnati	Original		< 5000 SQ.FT.
Dallas	Original		OVER 10000 SQ.FT.
Denver	Compact		< 5000 SQ.FT.
Los Angeles	Original		5000-9999 SQ.FT.
Miami	Compact		< 5000 SQ.FT.
Minneapolis	Compact		< 5000 SQ.FT.
Nashville	Original		< 5000 SQ.FT.
New Orleans	Original		OVER 10000 SQ.FT.
New York	Modern	demo\memo.doc	5000-9999 SQ.FT.
Philadelphia	Compact		OVER 10000 SQ.FT.
Phoenix	Original		< 5000 SQ.FT.
Pittsburgh	Compact		5000-9999 SQ.FT.
San Francisco	Original		5000-9999 SQ.FT.
Seattle	Compact		OVER 10000 SQ.FT.
St. Louis	Original		< 5000 SQ.FT.
Washington	Original		< 5000 SQ.FT.

4. From the **Tools** menu, select **Export Captions**.
5. Click **Browse** to select the location in which to write the output file, then click **OK**.
6. To export only new text strings and those that have been changed since the last export, select **Only export new or changed strings**.
7. To exclude the Description properties from the export, select **Exclude Descriptions**.
8. Click **OK**.

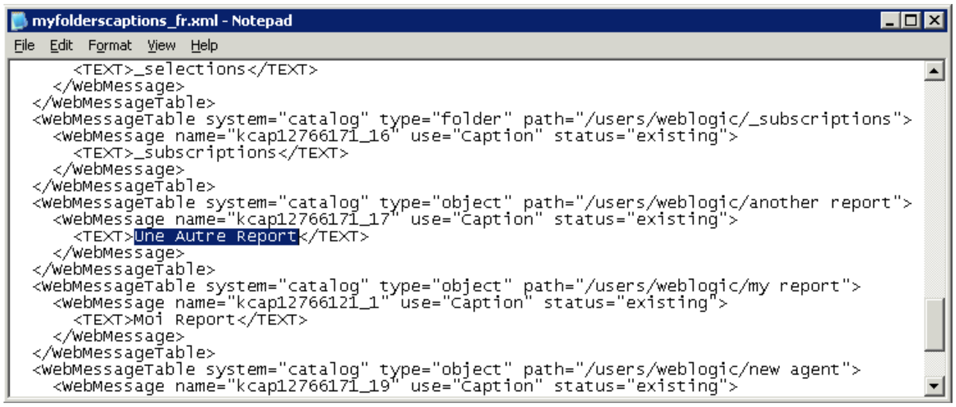
   The export process might take several minutes.
9. When the export process is complete, deliver the output file to the localization team.

   You can make a copy of every output file for each language to be translated.

   The following shows an extract from an exported caption XML file before translation (for example, myfoldercaptions.xml) and after translation (for example, myfoldercaptions_fr.xml).



```
<WebMessageTable system="catalog" type="folder" path="/users/weblogic/_selections">
  <WebMessage name="kcap12766171_15" use="Caption" status="existing">
    <TEXT>_selections</TEXT>
  </WebMessage>
</WebMessageTable>
<WebMessageTable system="catalog" type="folder" path="/users/weblogic/_subscriptions">
  <WebMessage name="kcap12766171_16" use="Caption" status="existing">
    <TEXT>_subscriptions</TEXT>
  </WebMessage>
</WebMessageTable>
<WebMessageTable system="catalog" type="object" path="/users/weblogic/another report">
  <WebMessage name="kcap12766171_17" use="Caption" status="existing">
    <TEXT>Another Report</TEXT>
  </WebMessage>
  <WebMessage name="kcap12766184_1" use="title" status="new">
    <TEXT>Another Report</TEXT>
  </WebMessage>
</WebMessageTable>
<WebMessageTable system="catalog" type="object" path="/users/weblogic/my report">
```



```
      <TEXT>_selections</TEXT>
    </webMessage>
  </webMessageTable>
  <webMessageTable system="catalog" type="folder" path="/users/weblogic/_subscriptions">
    <webMessage name="kcap12766171_16" use="Caption" status="existing">
      <TEXT>_subscriptions</TEXT>
    </webMessage>
  </webMessageTable>
  <webMessageTable system="catalog" type="object" path="/users/weblogic/another report">
    <webMessage name="kcap12766171_17" use="Caption" status="existing">
      <TEXT>Une Autre Report</TEXT>
    </webMessage>
  </webMessageTable>
  <webMessageTable system="catalog" type="object" path="/users/weblogic/my report">
    <webMessage name="kcap12766121_1" use="Caption" status="existing">
      <TEXT>Moi Report</TEXT>
    </webMessage>
  </webMessageTable>
  <webMessageTable system="catalog" type="object" path="/users/weblogic/new agent">
    <webMessage name="kcap12766171_19" use="Caption" status="existing">
```

To expose catalog text strings:

1. Place the translated XML files into their corresponding location in the following directory and restart Presentation Services:

   ORACLE_INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*\captions

   where *xx* is the language extension.

   For example:

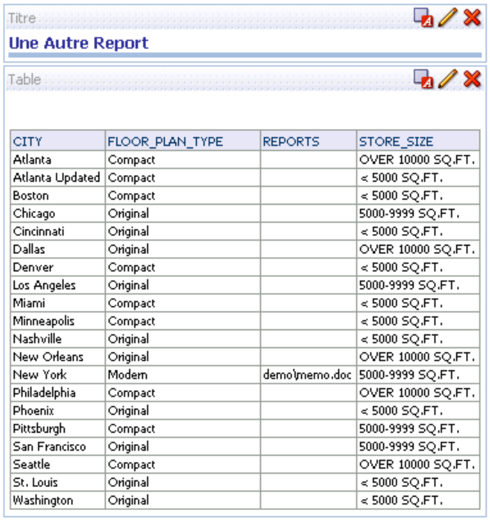
   D:\R11\instances\instance1\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips1\msgdb\l_fr\captions\myfoldercaptions_fr.xml

   Other examples of language extensions include cs for Czech and de for German.

2. Sign into Oracle Business Intelligence and select the appropriate language, such as French.

3. Display the translated content.

   For example, display the following translated title.



Une Autre Report

CITY	FLOOR_PLAN_TYPE	REPORTS	STORE_SIZE
Atlanta	Compact		OVER 10000 SQ.FT.
Atlanta Updated	Compact		< 5000 SQ.FT.
Boston	Compact		< 5000 SQ.FT.
Chicago	Original		5000-9999 SQ.FT.
Cincinnati	Original		< 5000 SQ.FT.
Dallas	Original		OVER 10000 SQ.FT.
Denver	Compact		< 5000 SQ.FT.
Los Angeles	Original		5000-9999 SQ.FT.
Miami	Compact		< 5000 SQ.FT.
Minneapolis	Compact		< 5000 SQ.FT.
Nashville	Original		< 5000 SQ.FT.
New Orleans	Original		OVER 10000 SQ.FT.
New York	Modern	demo\memo.doc	5000-9999 SQ.FT.
Philadelphia	Compact		OVER 10000 SQ.FT.
Phoenix	Original		< 5000 SQ.FT.
Pittsburgh	Compact		5000-9999 SQ.FT.
San Francisco	Original		5000-9999 SQ.FT.
Seattle	Compact		OVER 10000 SQ.FT.
St. Louis	Original		< 5000 SQ.FT.
Washington	Original		< 5000 SQ.FT.

To move translated captions from a development environment to a production environment:

- If the caption file:

- Does *not* exist in the production environment, then simply copy it from the development environment to the production environment.
- Does exist in the production environment, first make a backup copy of the existing file. Then open the caption file in the production environment in a text editor or XML editing tool and manually (and very carefully) insert the changes that were made in the development environment.

16.2.3 Tip for Arabic and Hebrew in Mozilla Firefox Browsers

By default, scroll bars are displayed on the right side of the Mozilla Firefox browser. If you are using the Arabic or Hebrew languages, then it is not appropriate to have the scroll bars on the right side. You can change the browser settings in Firefox such that the scroll bars are displayed on the left side.

For information on changing the layout.scrollbar.side setting, see the Firefox documentation.

16.2.4 Handling the Translation of Layers in Maps

You can use the functionality of MapViewer to label the features of a theme (called a layer for maps in Oracle BI EE) using a specific language or locale. To configure these translated labels for maps, use the information that is provided in *Oracle Fusion Middleware User's Guide for Oracle MapViewer*.

16.3 Setting the Current Locale in the Oracle BI Server

The following sections provide information about setting the locale in Oracle BI Server:

- Section 16.3.1, "Setting Locale Parameters on Oracle BI Server"
- Section 16.3.2, "Understanding How the Error Message Language is Determined"
- Section 16.3.3, "Troubleshooting the Current Locale in the Oracle BI Server"

16.3.1 Setting Locale Parameters on Oracle BI Server

To support multiple languages, the Oracle BI Server must be configured properly. The General section of the NQSConfig.INI file contains parameters that are required for localization and internationalization. It also contains default parameters that determine how data is sent from the Oracle BI Server to a client application. See Appendix A, "NQSConfig.INI File Configuration Settings" for complete information on these parameters.

The following parameters in the NQSConfig.INI file affect localization:

- LOCALE
- SORT_ORDER_LOCALE
- SORT_TYPE
- CASE_SENSITIVE_CHARACTER_COMPARISON

To successfully run Oracle Business Intelligence, ensure that you configure the appropriate locales on your operating system for the language in which users run the applications. Some locale- and language-related settings are interrelated and help determine how the Oracle BI Server sorts data.

16.3.1.1 Setting the Locale on UNIX Systems

The value to use for the C-run-time locale during server startup is specified in the SORT_ORDER_LOCALE parameter in the NQSConfig.INI file. This parameter is set normally by the Oracle BI Server. The locale is used for functions such as displaying dates and currencies and sorting data.

If you must adjust the setting, then in the General section of the NQSConfig.INI file, set the LOCALE and SORT_ORDER_LOCALE parameters, entering a platform-independent name as shown in Table 16–2.

Table 16–2 shows language mappings from the platform-independent name to the specific name for each of the supported UNIX platforms. For example, Chinese uses the setting zh_CN.utf8 on HP-UX or Linux operating systems.

Name strings such as zh_CN.utf8 and fr-FR-UTF-8 are the platform-specific names of the locale components, which must be installed by a system administrator. The NQSConfig.INI file uses the platform-*independent* names, such as Chinese or French (the names are case-insensitive).

Table 16–2 LOCALE Settings for UNIX Platforms

Locale (Platform-Independent Name)	Name on Solaris	Name on AIX	Name on HP-UX/Linux
Arabic	ar_SA.UTF-8	AR_AA.UTF-8	ar_SA.utf8
Chinese	zh_CN.UTF-8	ZH_CN.UTF-8	zh_CN.utf8
Chinese-traditional	zh_TW.UTF-8	ZH_TW.UTF-8	zh_TW.utf8
Croatian	hr_HR.UTF-8	HR_HR.UTF-8	hr_HR.utf8
Czech	cs_CZ.UTF-8	CS_CZ.UTF-8	cs_CZ.utf8
Danish	da_DK.UTF-8	DA_DK.UTF-8	da_DK.utf8
Dutch	nl_NL.UTF-8	NL_NL.UTF-8	nl_NL.utf8
English-USA	en_US.UTF-8	EN_US.UTF-8	en_US.utf8
Finnish	fi_FI.UTF-8	FI_FI.UTF-8	fi_FI.utf8
French	fr_FR.UTF-8	FR_FR.UTF-8	fr_FR.utf8
German	de_DE.UTF-8	DE_DE.UTF-8	de_DE.utf8
Greek	el_GR.UTF-8	EL_GR.UTF-8	el_GR.utf8
Hebrew	he_IL.UTF-8	HE_IL.UTF-8	iw_IL.utf8
Hungarian	hu_HU.UTF-8	HU_HU.UTF-8	hu_HU.utf8
Italian	it_IT.UTF-8	IT_IT.UTF-8	it_IT.utf8
Japanese	ja_JP.UTF-8	JA_JP.UTF-8	ja_JP.utf8
Korean	ko_KR.UTF-8	KO_KR.UTF-8	ko_KR.utf8
Norwegian	no_NO.UTF-8	NO_NO.UTF-8	no_NO.utf8
Polish	pl_PL.UTF-8	PL_PL.UTF-8	pl_PL.utf8
Portuguese	pt_PT.UTF-8	PT_PT.UTF-8	pt_PT.utf8
Portuguese-Brazilian	pt_BR.UTF-8	PT_BR.UTF-8	pt_BR.utf8
Romanian	ro_RO.UTF-8	RO_RO.UTF-8	ro_RO.utf8
Russian	ru_RU.UTF-8	RU_RU.UTF-8	ru_RU.utf8

Table 16–2 (Cont.) LOCALE Settings for UNIX Platforms

Locale (Platform-Independent Name)	Name on Solaris	Name on AIX	Name on HP-UX/Linux
Slovak	sk_SK.UTF-8	SK_SK.UTF-8	sk_SK.utf8
Spanish	es_ES.UTF-8	ES_ES.UTF-8	es_ES.utf8
Swedish	sv_SE.UTF-8	SV_SE.UTF-8	sv_SE.utf8
Thai	th_TH.UTF-8	TH_TH.UTF-8	th_TH.utf8
Turkish	tr_TR.UTF-8	TR_TR.UTF-8	tr_TR.utf8

16.3.2 Understanding How the Error Message Language is Determined

For Oracle BI Presentation Services, the error message language is set based on the NQ_SESSION.WEBLANGUAGE session variable. Presentation Services provides a default value for this variable upon installation. The value is updated when a user selects a language on the Oracle BI EE sign-in page.

For other clients, including third-party clients, the error message language is determined by the following precedence model:

- The error message language is set based on the WEBLANGUAGE session variable.
- If the WEBLANGUAGE session variable is not set, then the error message language is based on the error language that is specified in the ODBC Data Source Name (DSN) that is used to access the Oracle BI Server.

  See *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition* for information about setting the error message language in the ODBC DSN.
- If an error message language has not been set in the ODBC DSN, then the language that is specified in the ORACLE_BI_LANG environment variable is used for error messages.

  To change the value of ORACLE_BI_LANG, update the character code for this variable in opmn.xml. You can view the character codes for supported languages in the *ORACLE_HOME*/bifoundation/server/locale directory (for example, "en" for English, or "pt-BR" for Portuguese/Brazilian).
- If the ORACLE_BI_LANG environment variable is not set, then error messages are displayed in English.

Note that clients for the Administration Tool and Job Manager do not set the WEBLANGUAGE session variable. Because of this, these clients follow the precedence model starting with the ODBC DSN error message setting.

16.3.3 Troubleshooting the Current Locale in the Oracle BI Server

This section provides the following information on troubleshooting the current locale in the Oracle BI Server:

- Section 16.3.3.1, "Handling the NLS Locale Not Supported Error Message"
- Section 16.3.3.2, "Setting the Japanese Locale on AIX Systems"

16.3.3.1 Handling the NLS Locale Not Supported Error Message

If you do not have the appropriate locale installed, then the Oracle BI Server does not start, and the NQSServer.log file contains the following error:

[47013] NLS locale *xxx* is not supported by the operating system.

where *xxx* is the locale that is specified in the NQSConfig.INI file for the SORT_ORDER_LOCALE parameter. Take the following actions to resolve this error:

- **UNIX**. Install the locale that is indicated in Table 16–2 for the requested language.
- **Windows**. Add the corresponding language pack using the Regional Settings dialog box.

16.3.3.2 Setting the Japanese Locale on AIX Systems

When using a Japanese localization on an AIX platform, you might discover that the Oracle BI Server does not start. If you encounter this issue, then use the following procedure.

To set the Japanese locale on an AIX system:

1. Ensure that the JA_JP.UTF-8 locale is installed. If it is not, then install it.
2. Open the NQSConfig.INI file for editing, as described in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings."
3. In the General section, set the following parameters, being aware that the settings are case-sensitive:
   - LOCALE = "Japanese";
   - SORT_ORDER_LOCALE = "Japanese";
4. Save and close the NQSConfig.INI file.

16.4 Localizing Metadata Names in the Repository

You can use the Externalize Strings utility in the Administration Tool to localize the names of subject areas, tables, hierarchies, columns, and their descriptions in the Presentation layer. You can save these text strings to an external file with ANSI, Unicode, and UTF-8 encoding options.

To externalize strings for localization:

1. Open the repository in the Administration Tool.
2. Right-click any Presentation layer object, such as a subject area, presentation table, or presentation column, and select either **Externalize Display Names** then **Generate Custom Names**, or **Externalize Descriptions** then **Generate Custom Descriptions** to externalize strings.

   Selecting one of these right-click externalization options automatically selects the **Custom display name** or **Custom description** options in the Properties dialog for the selected object and all of its child objects.

   For example, if you right-click a subject area and select one of the externalization options, then the externalization flag is set on all presentation tables, columns, hierarchies, and levels within that subject area.
3. Select **Tools**, then select **Utilities**.
4. Select **Externalize Strings** and click **Execute**.
5. In the Externalize Strings dialog, select a subject area in the left pane.

   You can select all the subject areas at once, or select them individually and create a separate string file for each one.

In the right pane, the translated values and the original strings (names and descriptions) are displayed. These are placed in session variables for use by Presentation Services.

Only those objects with the externalization flag set in the Presentation layer are displayed in the right pane

6. Click **Save**.
7. In the Save As dialog, select a type of file and an encoding value and click **Save**.
8. In the Externalized Strings dialog, click **Close**.
9. (Optional) To disable externalization, right-click a Presentation layer object and select **Externalize Display Names**, then **Disable Externalization**, or **Externalize Descriptions** then **Disable Externalization**.

   Selecting one of these options automatically deselects the **Custom display name** or **Custom description** options in the Properties dialog for the selected object and all of its child objects.

When you have created the string file using the Externalize Strings utility, you can use it to translate the strings for the metadata objects, as described in the following procedure.

To translate strings for metadata from the exported string file:

1. Open the string file and examine the columns:
   - The first column contains the actual repository object names, which have a prefix of their type.
   - The second column contains the session variables that correspond to the name of each object or description, with a default prefix of CN_ for custom names and CD_ for custom descriptions.
2. In the third column of the file, ask the translation team to provide the translation of the name of each object.
3. Add a fourth column called Language. In this column, specify the code for the language in which the name was translated, such as de.
4. Load the string file into a database table.
5. In the Administration Tool, import the table into the physical layer.
6. Load the translated strings using row-wise initialization blocks. Ensure that you set the target of the initialization block to **Row-wise initialization** and that the execution precedence is set correctly.

   For example, you could do the following:

   a. Create a session initialization block that has the data source from a database, using a SQL statement such as the following one:

      ```
      SELECT 'VALUEOF(NQ_SESSION.WEBLANGUAGE)' FROM DUAL
      ```

   b. In the Session Variable Initialization Block dialog for SET Language, specify the LOCALE session variable for the Variable Target.

      This ensures that whenever a user signs in, the WEBLANGUAGE session variable is set. Then this variable sets the LOCALE variable using the initialization block.

c. Create another session initialization block that creates a set of session variables using a database-specific SQL statement such as the following one in the Session Variable Initialization Block Data Source dialog:

```
select SESSION_VARIABLE, TRANSLATION from external where LANGUAGE =
'VALUEOF(NQ_SESSION.LOCALE)'
```

This block creates all the variables whose language matches the language that the user specified during sign-in.

d. In the Session Variable Initialization Block Variable Target dialog, set the target of the initialization block to **Row-wise initialization**.

e. In the Execution Precedence area of the Session Variable Initialization Block dialog, specify the previously created initialization block, so that the first block that you created earlier is executed first.

7. Save your changes.

Tip: If you have an Oracle Application Development Framework data source, then you can propagate labels and tooltips from that data source, instead of using the Externalize Strings utility. See Appendix F, "Propagating Labels and Tooltips from Oracle ADF Data Sources" for more information.

16.5 Supporting Multilingual Data

This section describes how you can configure the Oracle BI Server to display field information in multiple languages, and contains the following topics:

- Section 16.5.1, "What is Multilingual Data Support?"
- Section 16.5.2, "What is Lookup?"
- Section 16.5.3, "What is Double Column Support?"
- Section 16.5.4, "Designing Translation Lookup Tables in a Multilingual Schema"
- Section 16.5.5, "Creating Logical Lookup Tables and Logical Lookup Columns"
- Section 16.5.6, "Creating Physical Lookup Tables and Physical Lookup Columns"
- Section 16.5.7, "Supporting Multilingual Data in Essbase Through Alias Tables"
- Section 16.5.8, "Enabling Lexicographical Sorting"

For information about using the Administration Tool, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

16.5.1 What is Multilingual Data Support?

Multilingual data support is the ability to display data from database schemas in multiple languages. Oracle BI Server supports multilingual schemas by simplifying the administration and improving query performance for translations. Multilingual schemas typically store translated fields in separate tables called lookup tables. Lookup tables contain translations for descriptor columns in several languages, while the base tables contain the data in the base language. Descriptor columns provide a textual description for a key column where there is a logical one-to-one relationship between the descriptor column and the key column. An example of a descriptor column might be Product_Name, which provides textual descriptions for a Product_Key column.

16.5.2 What is Lookup?

Lookup is when a query joins the base table and lookup table to obtain the translated values for each row in the base table.

Lookup tables might be dense and sparse in nature. A dense lookup table contains translations in all languages for every record in the base table. A sparse lookup table contains translations for only for some records in the base tables. Sometimes it is also possible that lookup tables are both dense and sparse. For example, a lookup table might contain complete translation for the Product Description field but only partial translation for the Product Category field. Dense and Sparse are types of lookup operation rather than being a table property. You configure lookup tables using the Administration Tool.

16.5.3 What is Double Column Support?

Double column support is the ability to associate two columns (a descriptor ID column and a descriptor column) in the logical layer, and can help you to define language independent filters. When the user creates a filter based on a descriptor column, the query tool displays a list of values to the user that are selected from the descriptor column.

This descriptor column technique is also useful when dealing with queries that involve LOB data types such as CLOBs and BLOBs and aggregate functions such as `COUNT` or `SUM`. Some data sources do not allow LOB columns to be used in the `GROUP BY` clause. So, instead of adding the LOB column to the `GROUP BY`, it is necessary to group by some other column that has a one-to-one relationship with the LOB column and then in join the LOB column after the aggregates have been computed.

16.5.4 Designing Translation Lookup Tables in a Multilingual Schema

There are two common techniques of designing translation lookup tables in a multilingual schema as follows:

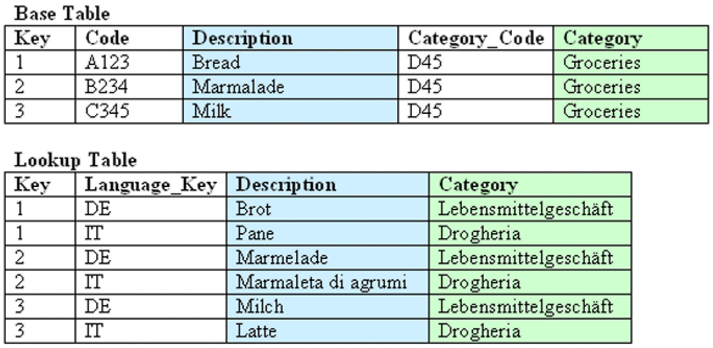
- Section 16.5.4.1, "A Lookup Table for Each Base Table"
- Section 16.5.4.2, "A Lookup Table for Each Translated Field"

16.5.4.1 A Lookup Table for Each Base Table

There is often a separate lookup table for each base table. The lookup table contains a foreign key reference to records in the base table, and contains the values for each translated field in a separate column. Assuming a completely dense lookup table, the number of rows in the lookup table for a particular language equals the number of rows in the base table.

The example in Figure 16–1 shows each record in the lookup table matching only one row in the base table.

Figure 16–1 A Lookup Table For Each Base Table



Base Table

Key	Code	Description	Category_Code	Category
1	A123	Bread	D45	Groceries
2	B234	Marmalade	D45	Groceries
3	C345	Milk	D45	Groceries

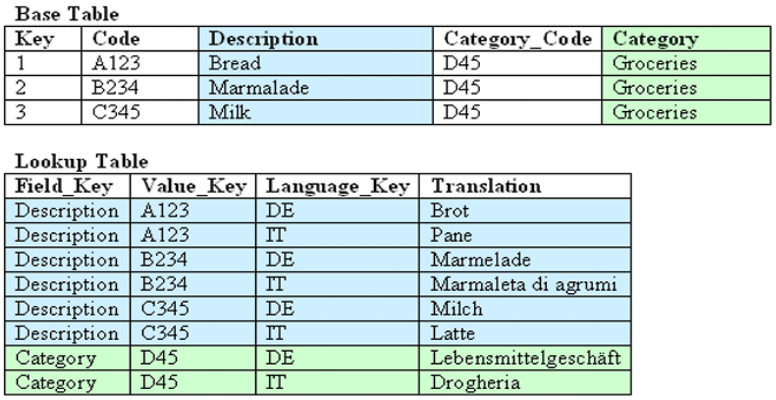
Lookup Table

Key	Language_Key	Description	Category
1	DE	Brot	Lebensmittelgeschäft
1	IT	Pane	Drogheria
2	DE	Marmelade	Lebensmittelgeschäft
2	IT	Marmaleta di agrumi	Drogheria
3	DE	Milch	Lebensmittelgeschäft
3	IT	Latte	Drogheria

16.5.4.2 A Lookup Table for Each Translated Field

The alternative approach to having one lookup table for each base table involves a separate lookup table for each translated field. Getting the translated value of each field requires a separate join to a lookup table. In practice there is often just one physical table that contains translations for multiple fields. When a single table contains translations for multiple fields, you must place a filter on the lookup table to restrict the data to only those values that are relevant to a particular column in the base table.

Figure 16–2 A Lookup Table For Each Translated Field



Base Table

Key	Code	Description	Category_Code	Category
1	A123	Bread	D45	Groceries
2	B234	Marmalade	D45	Groceries
3	C345	Milk	D45	Groceries

Lookup Table

Field_Key	Value_Key	Language_Key	Translation
Description	A123	DE	Brot
Description	A123	IT	Pane
Description	B234	DE	Marmelade
Description	B234	IT	Marmaleta di agrumi
Description	C345	DE	Milch
Description	C345	IT	Latte
Category	D45	DE	Lebensmittelgeschäft
Category	D45	IT	Drogheria

16.5.5 Creating Logical Lookup Tables and Logical Lookup Columns

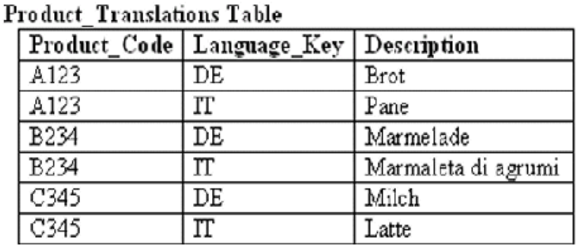
This section describes creating logical lookup tables and lookup columns and contains the following topics:

- Section 16.5.5.1, "Creating Logical Lookup Tables"
- Section 16.5.5.2, "Designating a Logical Table as a Lookup Table"
- Section 16.5.5.3, "About the LOOKUP Function Syntax"
- Section 16.5.5.4, "Creating Logical Lookup Columns"

16.5.5.1 Creating Logical Lookup Tables

You create a logical lookup table object in the business model to define the necessary metadata for a translation lookup table. A lookup table is a logical table with a property that designates it as a lookup table, as described in Section 16.5.5.2, "Designating a Logical Table as a Lookup Table."

Figure 16–3 Example of a lookup table



Product_Translations Table

Product_Code	Language_Key	Description
A123	DE	Brot
A123	IT	Pane
B234	DE	Marmelade
B234	IT	Marmaleta di agrumi
C345	DE	Milch
C345	IT	Latte

- Each of the lookup table's primary keys are considered together as a Lookup Key and perform the lookup. The lookup can be performed only when the values for all lookup key columns are provided. For example, in Figure 16–3, the combined Product_Code and Language_Key form the primary key of this lookup table.
- A lookup key is different from a logical table key because lookup key columns are order sensitive. For example, Product_Code and Language_Key are considered a different lookup key to Language_Key and Product_Code. You can specify the order of lookup key columns in the Administration Tool. All columns of the lookup key must be joined in the lookup function.
- A lookup table has only one lookup key.
- A lookup table has at least one value column. In Figure 16–3, the value column is Description, and it contains the translated value for the product description.
- There must be a functional dependency from a lookup key to each value column. In other words, the lookup key can identify the value column. The lookup key and value column should both belong to the same physical table.
- A lookup table is standalone without joining to any other logical tables.

  Consistency checking rules are relaxed for lookup tables, such that if a table is designated as a lookup table, it need not be joined with any other table in the subject area (logical tables would normally be joined to at least one table in the subject area).
- The aggregation results when using lookup columns should match the results from the base column. For example, the following code

  ```
  SELECT product.productname_trans, sales.revenue FROM snowflakesales;
  ```

  should return the same results as

  ```
  SELECT product.productname, sales.revenue FROM snowflakesales;
  ```

  If the lookup table productname_trans in this example uses the lookup key ProductID and LANGUAGE, then both queries return the same aggregation results.

  If the lookup key contains a column with a different aggregation level to productname, then the query grain changes and this affects the aggregation.

16.5.5.2 Designating a Logical Table as a Lookup Table

A logical table must be designated as a lookup table (using the Administration Tool) before you can use it as a lookup table. To designate a logical table as a lookup table, you must first import the lookup table into the physical layer and drop it into the Business Model and Mapping layer using the Administration Tool. Then, for each logical lookup table, you must select the **Lookup table** option in the Logical Table dialog box.

The order in which the columns are specified in the lookup table primary key determines the order of the corresponding arguments in the LOOKUP function.

For example, if the lookup table primary key consists of the RegionKey, CityKey, and LanguageKey columns, then the matching arguments in the LOOKUP function must be specified in the same order. You use the Administration Tool to change the order of primary key columns.

16.5.5.3 About the LOOKUP Function Syntax

A LOOKUP function is typically used in the Business Model and Mapping layer, as an expression in a translated logical table column.

The syntax of the LOOKUP function is as follows:

```
Lookup ::= LookUp([DENSE] value_column, expression_list ) | LookUp(SPARSE value_
column, base_column, expression_list )

expression_list ::= expr {, expression_list }

expr ::= logical_column | session_variable | literal
```

For example:

```
LOOKUP( SPARSE SnowflakeSales.ProductName_TRANS.ProductName,
SnowflakeSales.Product.ProductName, SnowflakeSales.Product.ProductID, VALUEOF(NQ_
SESSION."LANGUAGE"))

LOOKUP( DENSE SnowflakeSales.ProductName_TRANS.ProductName,
SnowflakeSales.Product.ProductID, VALUEOF(NQ_SESSION."LANGUAGE"))
```

Note the following:

- A LOOKUP function is either dense or sparse, and is specified using the keyword DENSE or SPARSE. The default behavior is dense lookup, if neither DENSE or SPARSE is specified. For DENSE lookup, the translation table is joined to the base table through an inner join, while for SPARSE lookup, a left outer join is performed.
- The first parameter (the parameter after the DENSE or SPARSE keyword) must be a valid value column from a valid lookup table that is defined in the logical layer.
- If the SPARSE keyword is given, then the second parameter must be a column that provides the base value of the value_column. For DENSE lookup, this base column is not required.
- The number of expressions in the expression_list should be equal to the number of the lookup key columns that are defined in the lookup table, which is defined by the value_column. The expression that is specified in the expression list should also match the lookup key columns one by one in order.

  For example:

- The lookup key for lookup table ProductName_TRANS is both Product_code and Language_Key
- The expressions in expression_list are SnowflakeSales.Product.ProductID and VALUEOF(NQ_SESSION."LANGUAGE")
- The meaning of the lookup is:

  return the translated value of ProductName from the translation table with the condition of Product_code = SnowflakeSales.Product.ProductID and Language_Key = VALUEOF(NQ_SESSION."LANGUAGE")

16.5.5.4 Creating Logical Lookup Columns

You use the Expression Builder in the Administration Tool to create a logical column that includes the lookup function. The value of the logical column depends on the language that is associated with the current user.

You create a new logical column using a derived column expression in the Column Source tab, for example to get the translated product name:

```
INDEXCOL( VALUEOF(NQ_SESSION."LAN_INT"), "Translated Lookup Tables"."Product".
"ProductName", LOOKUP( DENSE "Translated Lookup Tables"."Product Translations".
"ProductName", "Translated Lookup Tables"."Product"."ProductID",
VALUEOF(NQ_SESSION."WEBLANGUAGE")))
```

LAN_INT is a session variable that is populated by the session initialization block MLS and represents either the base language or other languages:

- 0 for base language (for example, en - English)
- 1 for other language codes (for example, fr - French, or cn - Chinese)

WEBLANGUAGE is a session variable that is initialized automatically, based on the language selected when a user logs in.

The INDEXCOL function helps to select the appropriate column. In the example above, the expression returns the value of the base column (ProductName) only if the user language is the base language (that is, when the value of session variable LAN_INT is 0). If the user language is not the base language (when the value of the session variable LAN_INT is 1), then the expression returns the value from the lookup table of the language that is passed in the WEBLANGUAGE session variable.

When you use the DENSE function (shown in the previous example), if there is no value for a column in the translated language, then the lookup function displays a blank entry.

When you use the SPARSE function (shown in the following example), and there is no value for a column in the translated language, then the lookup function displays a corresponding value in the base language.

```
INDEXCOL( VALUEOF(NQ_SESSION."LAN_INT"), "Translated Lookup Tables"."Product".
"ProductName", LOOKUP( SPARSE "Translated Lookup Tables"."Product Translations".
"ProductName", "Translated Lookup Tables"."Product"."ProductName", "Translated
Lookup Tables"."Product"."ProductID", VALUEOF(NQ_SESSION."WEBLANGUAGE")))
```

Note: You cannot use a derived logical column that is the result of a LOOKUP function as part of a primary logical level key. This limitation exists because the LOOKUP operation is applied after aggregates are computed, but level key columns must be available before the aggregates are computed because they define the granularity at which the aggregates are calculated.

You can use a derived logical column that is the result of a LOOKUP function as a secondary logical level key.

Handling Non-ISO Type Language Codes

If the data has non-ISO type language codes in the tables, then there should be a table that maps ISO language codes to non-ISO language codes. You can use the pre-existing WEBLANGUAGE variable that sets the ISO language code when a user logs in. You define a separate LANGUAGE variable whose initialization block runs a query against a mapping table to fetch the non-ISO language code filtered by the value from the WEBLANGUAGE variable. Table 16–3 provides a mapping table for non-ISO language codes. LANGUAGE is a non-ISO language code.

Table 16–3 Mapping Table for Non-ISO Language Codes

WEBLANGUAGE	LANGUAGE	LAN_INT
en	ENG	0
cn	CHI	1
fr	FRA	1

16.5.6 Creating Physical Lookup Tables and Physical Lookup Columns

You can create physical lookup table objects in the business model to define the necessary metadata for translation lookup tables. Physical lookup tables are similar to logical lookup tables in both semantics and usage. Physical lookup tables address the following scenarios that logical lookup tables cannot handle:

- The lookup table source is fragmented. In this case, use multiple physical lookup tables to hold the values. For example, translation values for fragmented product name data can be distributed in two physical lookup tables called productname_trans_AtoM and productname_trans_NtoZ.
- Different levels of translation tables are available. For example, translations are available in both an Essbase data source and a relational data source. It is preferable to use the same source as the base query.

Unlike logical lookup tables, which you designate by selecting an option in the Logical Table dialog box, you configure physical lookup tables by constructing lookup functions in the logical table source mapping.

For example, suppose that you have the following physical tables:

- A base table called Categories, with columns such as categoryid and categoryname.
- A translation table called Categories_Trans, with columns such as categoryid, language_key, and categoryname. The translated value of categoryname is determined through a combination of the categoryid and language_key columns.

Suppose that you have a logical table called Categories. In that table, you add a new logical column called categoryname_p, which is a translation column that depends on the current language. The column is not derived from any other logical column (unlike logical lookup columns).

The following procedure explains how to configure a physical lookup translation column using the previous example.

To configure a translation column that is derived from a physical lookup table:

1. Open the repository in the Administration Tool.
2. In the Business Model and Mapping layer, create a new logical column by right-clicking the appropriate logical table (for example, Categories) and selecting **New Object**, then **Logical Column**.
3. Provide a name for the logical column (for example, categoryname_p).
4. Select the Column Source tab.
5. In the **Logical Table Source** box under **Derived from physical mappings**, double-click the logical table source object that contains the base table column. The Column Mapping tab of the Logical Table Source dialog box is displayed.
6. Ensure that **Show unmapped columns** is selected.
7. In the **Expression** column for the new logical column (for example, categoryname_p), enter an expression such as the following:

   ```
   INDEXCOL(VALUEOF(NQ_SESSION."LAN_INT"),
   "DB_Name"."My_Category"."My_Schema"."Categories"."CategoryName", LOOKUP(SPARSE
   "DB_Name"."My_Category"."My_Schema"."CATEGORIES_TRANS"."CATEGORYNAME",
   "DB_Name"."My_Category"."My_Schema"."Categories"."CategoryName",
   "DB_Name"."My_Category"."My_Schema"."Categories"."CategoryID",
   VALUEOF(NQ_SESSION."LANGUAGE")))
   ```

   You can also use Expression Builder to create the expression.
8. Click **OK** in the Logical Table Source dialog box.
9. Click **OK** in the Logical Column dialog box.
10. Save your changes.

The Categories_trans physical translation table does not need to be incorporated into the logical table source. The INDEXCOL function checks that if the LAN_INT session variable is 0, then the categoryname column is fetched from the base table. Note the following about the LOOKUP function:

- The physical LOOKUP function works exactly the same as a logical LOOKUP function. The only difference is that all the references to logical tables and columns are replaced by physical tables and columns.
- The first column of the LOOKUP function is a value column, which is a translation value column from a translation table. The second column is the base value column, if a sparse lookup exists. The remaining columns are columns or values to be joined to the physical translation table, which is the table that is implied by the value column of the LOOKUP function.

Because you cannot use a dialog box to configure a physical lookup table, you must ensure that the order of the join columns and values is compatible with the column sequence that is displayed in the Physical Table dialog box for the physical translation table. For example, on the Keys tab of the Physical Table dialog box for the Categories_trans table, the primary key is composed of the CategoryID and Language_Key columns.

The columns that are specified in the LOOKUP function correspond to these columns:

- The following line:

  ```
  "DB_Name"."My_Category"."My_Schema"."Categories"."CategoryID"
  ```

  corresponds to the Categories_trans.CategoryID column.

- The following line:

  ```
  valueof(NQ_SESSION."LANGUAGE")
  ```

  corresponds to the Categories_trans.Language_key column.

See "Creating Logical Lookup Tables and Logical Lookup Columns" for information about lookup concepts like the LAN_INT and LANGUAGE session variables and full syntax information for the LOOKUP function.

16.5.7 Supporting Multilingual Data in Essbase Through Alias Tables

Often, members in Essbase cubes have separate aliases for each user language to enable users to view member names in their own language. Typically, you define a session variable to dynamically select the appropriate alias upon user login. See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about Essbase alias tables and how to use them with session variables.

16.5.8 Enabling Lexicographical Sorting

Lexicographical sorting is the ability to sort data in alphabetical order. Most data sources support lexicographical sorting. However, if you notice that lexicographical sorting is not working properly for a particular data source, then you can configure the Oracle BI Server to perform the sort rather than the back-end data source. To do this, ensure that ORDERBY_SUPPORTED is not selected in the Features tab of the Database dialog box in the Administration Tool. See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information on specifying database features.

Note that disabling ORDERBY_SUPPORTED in the data source can have a very large performance impact, because consequently, many joins are not pushed down to the data source. In many cases, the performance impact is significant enough that ORDERBY_SUPPORTED should still be enabled in the data source, regardless of the impact on the lexicographical sorting functionality.

17

Configuring Currency Options

When content designers create analyses, they often include data that shows currency, such as American dollars. You can perform the following tasks that affect currency options:

- Section 17.1, "Changing the Default Currency for Analyses"
- Section 17.2, "Defining User-Preferred Currency Options"

17.1 Changing the Default Currency for Analyses

You can change the default currency that is displayed, for example, from French Francs to Euros. For information about using formatting functions in Answers, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

To set the default currency:

1. Open the currencies.xml file in the directory *ORACLE_HOME*\bifoundation\web\display.
2. Search for the currency to make the default, for example, USD, CAD, PEN, or MAD.
3. Copy the entire currency element.

   For example, copy the currency tag for the Euro:

   ```
   - <Currency tag="int:euro-l" type="international" symbol="_"
   displayMessage="kmsgCurrencyEuroLeft" digits="2" format="$ #">
   <negative tag="minus" format="-$ #" />
   </Currency>
   ```

4. Search for the text string int:wrhs, located near the top of the file.
5. Select the entire element and replace it by pasting the copied element over it.
6. Replace the tag attribute so it reads int:wrhs.

   For example, replace `tag="int:euro-l"` with `tag="int:wrhs"`.

7. Restart the service for Oracle BI Presentation Services.

To specify the currency for a column in a customized subject area:

1. In Answers, modify the analysis that uses the subject area.
2. In the Analysis editor: Criteria tab, click the **Options** button for the currency column and select **Column Properties** to display the Column Properties dialog.
3. Click the Data Format tab and select the **Override Default Data Format** box.

4. In the **Treat Numbers As** box, select **Currency**.
5. In the **Currency Symbol** list, select the symbol.
6. Complete the other options on the tab as appropriate.
7. If desired, save this setting as a systemwide default.
8. Click **OK** when you have finished, and repeat the preceding steps for any other columns to change.

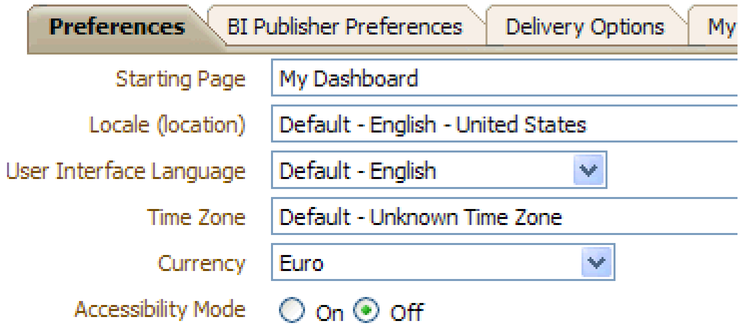
> **Note:** If you are using Oracle Business Intelligence applications, with no customizations, then you must set only the default data warehouse currency.
>
> If you have created additional subject areas, then the currency column data is in the number format, and you must specify the currency for the customized subject area as described in the second procedure.

17.2 Defining User-Preferred Currency Options

Users can select the currency in which they prefer to view currency columns in analyses and dashboards in two ways:

- In the **Currency** box on the My Account dialog: Preferences tab
- In currency prompts

Figure 17–1 Example of the Currency box on the My Account dialog: Preferences tab



Starting Page: My Dashboard

Locale (location): Default - English - United States

User Interface Language: Default - English

Time Zone: Default - Unknown Time Zone

Currency: Euro

Accessibility Mode: ○ On ● Off

For information about setting the currency preference on the My Account dialog: Preferences tab or about currency prompts, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

You define the currency options that are to be displayed in the **Currency** box and in a currency prompt in the userpref_currencies.xml file. (These currency options must be for currencies to which your installation can convert columns.) Defining the currency options also controls whether the **Currency** box is available on the My Account dialog: Preferences tab and whether the **Currency Prompt** option is available on the Definition pane of the Prompt editor.

When you define these currency options, you can use one of two types of mappings:

- **Static** — Produces a static list of currency options that all users see.
- **Dynamic** — Produces a list of currency options that changes dynamically based on a logical SQL statement that you specify. This is useful, for example, to dynamically change the currency options based on the user.

Note: In order for the user-preferred currency options to take effect, the following configuration also must be done in the Oracle Business Intelligence repository:

- Creation of the PREFERRED_CURRENCY session variable
- Conversion setup of logical currency columns in the Business Model and Mapping layer
- Creation of the userCurrencyPreference table using the currency information from your installation that enables you to dynamically change the currency options based on a logical SQL statement that you specify (required only if you use a dynamic mapping)

For information, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

17.2.1 Defining User-Preferred Currency Options Using a Static Mapping

To define the user-preferred currency options using a static mapping:

1. Use a text editor to open the userpref_currencies.xml file located in the following directory:

   ORACLE_INSTANCE\config\OracleBIPresentationServicesComponent\coreapplication_obips*n*

2. Add a UserCurrencyPreferences element as follows:

   ```
   <UserCurrencyPreferences currencyTagMappingType="static">

   </UserCurrencyPreferences>
   ```

3. For each currency option to be displayed in the **Currency** box or in currency prompts, add a UserCurrencyPreference element between the <UserCurrencyPreferences> tags using this format:

   ```
   <UserCurrencyPreference sessionVarValue="sessionVarValuevalue"
   displayMessage="displayMessagevalue" displayText="displayTextvalue"
   currencyTag="currencyTagvalue"/>
   ```

   where:

   - sessionVarValue="*sessionVarValue* " sets the session variable PREFERRED_CURRENCY. For its value, specify a string that uniquely identifies the currency, for example, `gc1`.
   - (optional) displayMessage="*displayMessagevalue*" sets the presentation variable currency.userPreference to a localized value. To specify a localized value, you first must create the localized message for the currency in the usercurrencymessages.xml file. For information, see Section 16.2.1.2, "Localizing Messages for Users' Preferred Currency." Then, for the value of displayMessage, specify the WebMessage name that is identified in the usercurrencymessages.xml file for the currency. For example, if you created this English language entry in the usercurrencymessages.xml file:

     ```
     <WebMessage name="kmsgMyCurrency1"><TEXT>My Currency 1</TEXT></WebMessage>
     ```

     Then you would specify kmsgMyCurrency1 as the value of displayMessage.

- (optional) displayText="*displayTextvalue*" sets the presentation variable currency.userPreference to a value that is not localized. For its value, specify a string that identifies the currency, such as Global Currency 2.

  For more information about the currency.userPreference variable, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*

- currencyTag="*currencyTagvalue*" identifies the Currency Tag in the currencies.xml file whose displayMessage value is to be used to populate the Currency box on the My Account dialog: Preferences tab and currency prompts. (The currencies.xml file, which is located in ORACLE_HOME\bifoundation\web\display, provides currency formats.)

  Note: The value of the currency.userPreference variable is obtained from the displayMessage and displayText attributes of the UserCurrencyPreference element using the following order of precedence:

  1. displayText
  2. displayMessage

  If no values exist for displayText and displayMessage, then the value of the displayMessage attribute for the corresponding currency tag in the currencies.xml file is used.

4. Save and close the userpref_currencies.xml file.
5. Restart Oracle Business Intelligence.

   For information, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

17.2.2 Example: Static Mapping to Define User-Preferred Currency Options

The following example shows a userpref_currencies.xml file that uses a static mapping to define user-preferred currency options:

```
<UserCurrencyPreferences currencyTagMappingType="static">
<UserCurrencyPreference sessionVarValue="none" displayText="None"/>
<UserCurrencyPreference sessionVarValue="gc1" displayMessage="kmsgGlobalCurrency1"
currencyTag="int:USD"/>
<UserCurrencyPreference sessionVarValue="gc2" displayText="Global Currency 2"
currencyTag="int:euro-l"/>
<UserCurrencyPreference sessionVarValue="gc3" displayText="Global Currency 3"
currencyTag="int:USD"/>
<UserCurrencyPreference sessionVarValue="orgc" displayText="Org Currency"
currencyTag="int:USD"/>
<UserCurrencyPreference sessionVarValue="lc1" displayTag="int:DEM"
currencyTag="int:USD"/>
</UserCurrencyPreferences>
```

17.2.3 Defining User-Preferred Currency Options Using a Dynamic Mapping

To define user-preferred currency options using a dynamic mapping:

1. Use a text editor to open the userpref_currencies.xml file that is located in the following directory:

ORACLE_INSTANCE\config\OracleBIPresentationServicesComponent\coreapplication_obips*n*

2. Add a UserCurrencyPreferences element as follows:

```
<UserCurrencyPreferences currencyTagMappingType="dynamic">

</UserCurrencyPreferences>
```

3. Add a UserPrefCurrencyLogicalSQL element between the <UserCurrencyPreferences> tags using this format:

```
<UserPrefCurrencyLogicalSQL>
SELECT column1, column2, column3 FROM userCurrencyPreference
</UserPrefCurrencyLogicalSQL>
```

where:

- *column1* contains the values that are used to set the session variable PREFERRED_CURRENCY. Each value in this column is a string that uniquely identifies the currency, for example, `gc1`.
- *column2* contains the currency tags in the currencies.xml file whose displayMessage values are to be used to populate the Currency box and currency prompts, for example, `int:euro-1`. (The currencies.xml file, which is located in ORACLE_HOME\bifoundation\web\display, provides currency formats.)
- (optional) *column3* contains the values used to set the presentation variable currency.userPreference. Each value in this column is a string that identifies the currency, such as Global Currency 2.

Note: If you omit *column3*, then the values for the displayMessage attributes for the corresponding currency tags in the currencies.xml file are used.

For more information about the currency.userPreference variable, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*

4. Save and close the userpref_currencies.xml file.
5. Restart Oracle Business Intelligence.

For information, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

17.2.4 Example: Dynamic Mapping to Define User-Preferred Currency Options

The following example shows a userpref_currencies.xml file that uses a dynamic mapping to define user-preferred currency options:

```
<UserCurrencyPreferences currencyTagMappingType="dynamic">
UserPrefCurrencyLogicalSQL>
<!-- In this SELECT statement, column1 contains the values to set the PREFERRED_
CURRENCY variable, column2 contains the currency tag values, and column3 contains
the values to set the currency.userPreference variable. -->
SELECT markets.userpreferences, markets.currencyTag, markets.userpreferencename
FROM userCurrencyPreference
</UserPrefCurrencyLogicalSQL>
```

```
</UserCurrencyPreferences>
```

Table 17–1 shows sample results from the logical SQL statement.

Table 17–1 Sample Logical SQL Results

"Markets"."UserPreference"	"Markets"."CurrencyTag"	"Markets"."UserPreferenceName"
char	char	char
orgc1	loc:en-BZ	Org currency
gc2	int:euro-1	Global currency 2
lc1	int:DEM	Ledger currency
gc1	int:USD	Global Currency 1

18

Configuring and Managing the Oracle BI Presentation Catalog

This chapter describes how to configure and manage the Oracle BI Presentation Catalog and provides information about basic maintenance procedures and configuring for full-text searching.

This chapter contains the following sections:

18.1 About the Oracle BI Presentation Catalog

The Oracle BI Presentation Catalog stores the content that users create in a directory structure of individual files. This content includes folders, shortcuts, Oracle BI EE objects (such as analyses, filters, prompts, and dashboards), and Oracle BI Publisher objects (such as reports and templates).

Each object is stored in its own file. For example, an analysis called Analysis 1 would be stored in a file named Analysis1. Each object has a corresponding attributes file. For example, the analysis called Analysis1 would have a corresponding attributes file named Analysis1.atr. The attributes file contains the object's full name, access control list (ACL), description, and so on. To access an object in the catalog, users must have appropriate ACL entries for that object. All objects in the catalog use ACL entries.

> **Note:** To guarantee that only one user can write to a file at one time, a lock file is created when an object is being written to. On rare occasions (for example, after a power outage), temporary lock files in the Oracle BI Presentation Catalog might not get cleaned up completely. If Presentation Services reports of such a lock file, then you must delete it manually.

The following locations are the default locations for the catalog directory:

- On Windows

  ORACLE_INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\catalog

- On UNIX

  ORACLE_INSTANCE/bifoundation/OracleBIPresentationServicesComponent/coreapplication_obips*n*/catalog

18.2 About Catalog Manager

Catalog Manager is a tool that lets you perform online and offline management of Oracle BI Presentation Catalogs. It should be installed on a secure computer that is accessible only to Oracle BI Administrators.

18.2.1 Uses for Catalog Manager

You can use Catalog Manager to:

- Manage folders, shortcuts, and objects (analyses, filters, prompts, dashboards, and so on). For example, you can rename and delete objects, and you can move and copy objects within and between catalogs.
- View and edit catalog objects in Extensible Markup Language (XML).
- Preview objects, such as analyses and prompts.
- Search for and replace catalog text.
- Search for catalog objects.
- Create analyses to display catalog data.
- Localize captions. See Section 16.2.2, "Localizing Oracle BI Presentation Catalog Captions."

Many of the operations that you can perform in Catalog Manager, can also be performed through the Catalog page in Oracle BI Presentation Services. For information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

18.2.2 Guidelines for Working with Catalog Manager

Follow these guidelines when working with Catalog Manager:

- Always make backup copies of the Oracle BI Presentation Catalogs that you are working with.

- Be sure of changes that you plan to make. Catalog Manager commits changes immediately. There is no undo function nor are there any error messages to tell you that a particular change does not display well to users. However, if you do make any unwanted changes, then you can revert to your latest saved backup.
- Do not copy and paste into e-mail, as this is not supported.

18.2.3 Tips for Working with Catalog Manager

As you work with Catalog Manager, keep the following tips in mind:

- While working in online mode, you can paste catalog contents into or out of a read-only folder by turning off the read-only property of the folder tree before copying, then re-apply the read-only attribute after pasting.
- You cannot copy, archive, or drag files from the /system/security directory in the Catalog Manager.
- Some keyboard shortcuts might not work properly.
- Even if a resize indicator is not shown, Catalog Manager panes might still be resizable.

18.3 Maintaining the Oracle BI Presentation Catalog

This section contains the following topics on maintaining a catalog:

- Section 18.3.1, "Manually Changing Additional Configuration Settings for the Catalog"
- Section 18.3.2, "Manually Creating a New Oracle BI Presentation Catalog"
- Section 18.3.3, "Deploying a Catalog Object to Production"

18.3.1 Manually Changing Additional Configuration Settings for the Catalog

In addition to the presentation settings that you can change in Fusion Middleware Control, you can modify other settings manually. Use various elements in the instanceconfig.xml file to change these settings.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually change additional configuration settings for the catalog:

1. Open the instanceconfig.xml file for editing as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the following element:
   - **HashUserHomeDirectories**: Specifies the hashing of directories. If you have more than 4000 catalog users or if you intend to have more than 4000 catalog users in the future, then you must turn on the hashing of users' home directories to address a file system limitation. To do so, set the HashUserHomeDirectories element to 2 from its default value of 0. When this element is turned on, for example, the default name for user Steve's home directory would become /users/st/steve.

Caution: The HashUserHomeDirectories element must be set immediately after installing Oracle BI EE to be effective.

3. Include the element and its ancestor elements as appropriate, as shown in the following example:

   ```
   <ServerInstance>
   <Catalog>
       <HashUserHomeDirectories>2</HashUserHomeDirectories>
   </Catalog>
   </ServerInstance>
   ```

4. Save your changes and close the file.
5. Restart Presentation Services.

18.3.2 Manually Creating a New Oracle BI Presentation Catalog

You can manually create a new catalog.

To manually create a new catalog:

1. Stop the service for Presentation Services.

   The Oracle BI Server and WebLogic Server must be running.

2. Specify a new location (one that does not exist) for the catalog on the Repository tab of the Deployment page in Fusion Middleware Control.

   See Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location."

3. Ensure that the new location for the catalog is empty.
4. Save your changes and close the file.
5. Restart Presentation Services.

18.3.3 Deploying a Catalog Object to Production

You can deploy a simple object (for example, a dashboard with privileges) to your production environment using the archive and unarchive functionality.

Deploying a complex object (for example, an object that includes references to external filters) is a more advanced process. Oracle offers advanced training courses to guide you through this process. It is recommended that you attend one of these courses before you attempt to deploy a complex object to production.

To deploy a catalog object to a production environment:

1. (Optional) If you are deploying a catalog object to a new production environment.

   Archive the catalog object in the test environment and unarchive it in the production environment as follows:

   a. Archive the catalog object in the test environment using one of the following:

      – Oracle BI Presentation Services.

      – Catalog Manager.

        For information, see Section 18.10, "Archiving and Unarchiving a Catalog Using Catalog Manager."

   b. Copy the archived file to the production computer.

   c. On the production computer, unarchive the object.

      For information on how to unarchive an object, see Section 18.10, "Archiving and Unarchiving a Catalog Using Catalog Manager."

   d. Set the permissions on the object as appropriate.

2. (Optional) If you are deploying the catalog to an existing production environment.

   Copy and paste new or updated objects from the test catalog into the production catalog as follows:

   a. Open two Catalog Manager windows: one with the test catalog, and another with the production Catalog.

   b. Selectively copy and paste the folders from the test catalog into the production catalog.

      If you copy and paste folders where the same content has been changed in the test or production environments, then test content overwrites the production content.

3. (Optional) If you are deploying the catalog from test to a new production environment, use Fusion Middleware Control to specify the location of the new catalog in the production environment.

   For more information, see Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location."

18.4 Starting Catalog Manager and Opening Catalogs

This section describes the following topics:

- Section 18.4.1, "Starting Catalog Manager"
- Section 18.4.2, "Understanding the Two Catalog Modes"
- Section 18.4.3, "Operations Available in Online Mode and Offline Mode"
- Section 18.4.4, "Opening an Oracle BI Presentation Catalog"

18.4.1 Starting Catalog Manager

Use the following procedure to start Catalog Manager.

To start Catalog Manager:

1. On Windows, double-click the runcat.cmd script in the following directory:

   ORACLE_HOME\bifoundation\web\catalogmanager

   or

   Using the command line, provide the full path to the script:

   runcat.cmd (on Windows)

   runcat.sh (on UNIX)

18.4.2 Understanding the Two Catalog Modes

You can open a catalog in one of two modes:

- Online mode — Connects to a catalog on a running Web server. In this mode your permissions are applied. You can see only those objects for which you have the appropriate permissions. Both Presentation Services and the Web Server must be running for you to open catalogs in online mode.

  Use online mode when you want to make minor incremental changes or additions to the catalog, such as changes to permissions, updates to a single object, or migration of new objects to a production environment

- Offline mode — Connects to a local file system. In this mode, you are logged in as a super user or system user, and no permissions are applied. You can see all objects in the catalog.

  Use offline mode when you want to make catalog-wide changes, such as globally renaming objects or moving multiple objects for reorganization.

Note: Generally, working in offline mode is faster than working in online mode. This is because you are accessing, creating, and updating the individual files directly, and the catalog does not have to communicate with Presentation Services as it does when you are working in online mode.

18.4.3 Operations Available in Online Mode and Offline Mode

Many of the operations that you can perform using Catalog Manager are available in both online mode and offline mode. A few operations are available in only one mode or the other. Generally, the operations available in:

- Online mode are read-only operations and write operations that do not affect the entire catalog, such as setting permissions for an object.
- Offline mode include most of the operations available in online mode and write functions that affect the entire catalog, such as searching for and replacing catalog text.

You can perform the following operations in online and offline modes (or as stated), as follows:

- Cutting objects
- Copying objects
- Pasting objects
- Copying objects for another catalog
- Pasting objects from another catalog
- Creating shortcuts of objects
- Deleting objects
- Renaming objects without reference updates
- Refreshing the Catalog Manager workspace
- Creating folders
- Setting permissions for objects
- Working with properties of objects
- Managing the view of the workspace

- Searching for objects
- Creating reports to display Catalog Manager data
- Setting browser preference
- Either:
  - Previewing objects (available in online mode only)
  - Renaming objects with reference updates (known as Smart Rename and available in offline mode only)
- Exporting captions for localization purposes
- Searching for and replacing catalog text (available in offline mode only

18.4.4 Opening an Oracle BI Presentation Catalog

To open an Oracle BI Presentation Catalog:

1. In Catalog Manager, from the **File** menu, select **Open Catalog**.
2. Complete the necessary fields.

   Some fields are described in the following list.

   - **Type** — Select the mode (online or offline) in which to open the catalog
   - **Path** — If you are opening the catalog in offline mode, then enter the path to the catalog folder on the local file system, for example:

     Click **Browse** to display a dialog for locating the catalog.

     C:\ORACLE_INSTANCE\\bifoundation\\OracleBIPresentationServicesComponent\\coreapplication_obips*n*\\catalog\\default
   - **URL** — If you are opening the catalog in online mode, then enter the URL to Oracle BI Presentation Services, for example:

     http://<hostname>/analytics/saw.dll
   - **User** — If you are opening the catalog in online mode, then enter the user name for the host URL (disabled in offline mode).
   - **Password** — If you are opening the catalog in online mode, then enter the password for the host URL (disabled in offline mode).
   - **Read-Only** — Select this field to open the catalog in read-only mode (disabled in offline mode).
3. Click **OK**.

18.5 Using the Catalog Manager Workspace

This section provides the following topics on the workspace for Catalog Manager:

- Section 18.5.1, "What Does the Catalog Manager Workspace Do?"
- Section 18.5.2, "What Does the Catalog Manager Workspace Look Like?"
- Section 18.5.3, "Managing the View of the Catalog Manager Workspace"

18.5.1 What Does the Catalog Manager Workspace Do?

The Catalog Manager workspace enables you to view and work with catalog objects. It displays the following folders for an open catalog:

- The shared folder— Contains content that is shared among catalog users. This includes the preconfigured dashboards and analyses that are distributed with prebuilt applications, and other objects such as shared filters.
- The system folder — Contains administrative elements of Presentation Services. Some of these elements are distributed with the product, and others are set up by you as the administrator, such as privileges.
- The users folder — Contains content that catalog users with the appropriate permissions have saved to their personal folders, such as individual analyses.

18.5.2 What Does the Catalog Manager Workspace Look Like?

Catalog Manager consists of the following main components:

- Menu bar — Lets you access the following menus:
  - File — Provides options that let you open and close catalogs, exit Catalog Manager, and so on.
  - Edit — Provides options that let you manage catalog objects, such as Cut, Copy, Permissions, and so on. (Many of these options are also available on the right-mouse pop-up menu.)
  - View — Provides options to manage the view of the Catalog Manager workspace, such as **Show Tree**, **Show Job Status**, and so on.
  - Tools — Provides options that let you manage catalogs, such as **XML Search and Replace**, **Create Report**, and so on.
  - Help — Provides options to access the Oracle BI Enterprise Edition Web site and to display information about Catalog Manager.
- Toolbar — Provides quick access to commonly used options, such as Cut, Copy, Paste, and so on.
- Tree pane — Displays catalog folders. The pane also displays objects but only if the **Show Objects in Tree** option on the View menu is selected.
- Table pane — Displays catalog folders and objects. It consists of:
  - The navigation bar, where you can move to the catalog object to work with by typing its path name.
  - These columns: Name, Type, Owner, My Permissions, Attributes, Date Created, and Last Modified. Click the column name to sort by that value, such as by type.

    The Type column identifies the type of object. Objects that are identified as "unknown file" are generally internally-used objects, and their type is not exposed in Catalog Manager.
- Right-mouse pop-up menu — Provides options that let you manage catalog objects, such as Rename, Properties, Permissions, and so on. (Many of these options are also available on the Edit menu.)

18.5.3 Managing the View of the Catalog Manager Workspace

You can manage what you view in the Catalog Manager. For example, you can show objects in the Tree pane or show job statuses.

To manage the view of the Catalog Manager workspace:

1. In Catalog Manager, choose **View** and then one of these options:
    - **Show Tree** — Displays the Tree pane, if you previously had closed it.
    - **Show Table** — Displays the Table pane, if you previously had closed it.
    - **Show Job Status** — Displays the Background Job Status pane, where you can view the progress of processes that you have run, such as Search and Replace, Smart Rename, and so on. You can also remove all finished jobs and set progress preferences using the icons in the upper-right corner of the pane.
    - **Show Objects in Tree** — Displays objects (that is, analyses, filters, and so on) in addition to folders in the Tree pane.
    - **Refresh** — Refreshes the objects that are displayed in the Tree and Table panes. (You might want to refresh the data, for example, if someone else makes changes to the catalog while you are working with it and you want to see the changes.)

18.6 Working with Objects in Catalog Manager

This section provides the following information on working with objects:

- Section 18.6.1, "Searching for Catalog Objects Using Catalog Manager"
- Section 18.6.2, "Copying and Pasting Objects"
- Section 18.6.3, "Renaming Catalog Objects"
- Section 18.6.4, "Working with the Properties of Catalog Objects"
- Section 18.6.6, "Previewing Objects from Catalog Manager"
- Section 18.6.5, "Setting Permissions of Catalog Objects"

In the Catalog page of Presentation Services, you can view folders and contents including hidden objects. You can create, rename, copy, move, and delete folders and contents. For more information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

> **Note:** Changes made in the Presentation layer of the Oracle BI Administration Tool can affect analyses and dashboards based on those tables and columns. You can use Catalog Manager to keep the catalog synchronized with these changes in the Presentation layer.

18.6.1 Searching for Catalog Objects Using Catalog Manager

You can search for objects in the catalog using the Search function. For example, you might want to search for all objects that have a property with the value of "administrator."

When you search, you can limit the search by:

- Case Sensitive — Select this check box to apply case sensitivity to the search criteria. The default value is unchecked.

- Name — Limits the search to the names of objects.
- Description — Limits the search to the Description property.
- Property values — Limits the search to the values of properties.
- Owner — Limits the search to the owners of objects.
- Object type — Searches for all types of objects or limits the search to a specific type of object that you specify (for example, analyses, filters, agents, dashboard prompts, dashboard pages).
- Date — Limits the search to objects that were created on the dates that you specify or to objects that were last modified on the dates that you specify.

To search for an object:

1. In Catalog Manager, open the catalog and navigate to the location in the tree where you want to begin the search.
2. Click **Search** on the toolbar.
3. In the **Search for any or all criteria below** field, enter the word or phrase to search for.
4. To make the search case sensitive, select the **Case Sensitive** box.
5. To limit the search, then click **Advanced Search**.
6. In the Advanced Search area, specify the constraints for the search.
7. Click **Search**.

> **Tip:** When you have finished searching, click **Explore the entire catalog tree** on the toolbar to return to the Tree and Table panes.

18.6.2 Copying and Pasting Objects

You can copy and paste objects within a single catalog. You can also copy objects from one catalog and paste them into another catalog.

18.6.2.1 Tips for Copying and Pasting

Use the following tips as you copy and paste objects:

- You can copy and paste objects using menu options, and you can use drag and drop functionality to copy objects between two catalogs and within the same catalog. Drag and drop always makes a copy of the dragged objects, even when performing a drag and drop within a single catalog.
- Catalogs are structured in hierarchical folders. When copying or merging objects, remember to also copy any objects that are associated with them, such as dashboard folders, shortcuts, and analyses. URL paths in external applications can be reestablished after a copy or merge operation if the entire folder path is not copied, for example, if added to the dashboard as a shortcut or text.
- Most often, you can simply copy and paste objects as needed. If required, you can set advanced options that affect the objects that you are pasting. For complete information, see Section 18.6.2.4, "Advanced Options for Pasting Objects."

18.6.2.2 Online and Offline Mode for Copying and Pasting

When you are copying between catalogs, the mode in which the catalogs are opened makes no difference. You can copy and paste objects from an:

- Online catalog to an offline catalog (or an offline catalog to an online catalog)
- Online catalog to another online catalog
- Offline catalog to another offline catalog

This is especially useful, for example, to develop new objects in a catalog offline and migrate them to a production environment without having to shut down Presentation Services.

18.6.2.3 Copying and Pasting Objects Between Catalogs

The following procedure describes how to copy and paste objects between two catalogs using menu options. If the two catalogs have the same name, then you might want to rename one of the catalogs before opening it to help distinguish between the two catalogs as you work. Both catalogs must be the same version 11.1.1 (or later).

To copy and paste objects between catalogs using menus:

1. In Catalog Manager, open the catalog to change, that is the target catalog.
2. Using another instance of the Catalog Manager, open the catalog that contains the objects to copy, that is the source catalog.
3. If necessary, reposition both instances of Catalog Manager on your screen so you can display the title bars of both Catalog Manager instances.
4. In the source catalog, right-click the source object and select **Copy**.
5. In the target catalog, right-click at the point where you want to paste the source object and select **Paste**.

18.6.2.4 Advanced Options for Pasting Objects

You can set advanced options for pasting objects that you have copied. These options are available in the Preferences dialog and must be set before the copy and paste operation begins.

18.6.2.4.1 Paste Overwrite The Preferences dialog contains the following options in the Paste Overwrite area:

- **Force** — Pastes all files, even those that have the read-only attribute set.
- **All** — Pastes all files, whether they exist or not. (Default)
- **Old** — Pastes all files, but does not paste any existing files unless they are older than the source.
- **None** — Pastes all files, but does not paste any existing files.

Consider the following example of pasting with overwrite options set. Suppose that the /users/joe folder contains the following analyses:

Analysis A (created 01-Jan-2010)
Analysis B (created 31-May-2010)
Analysis C (created 01-Jan-2010)

Suppose that the /users/sue folder contains the following analyses, but no Analysis C

Analysis A (created 28-Feb-2010)
Analysis B (created 01-Jan-2010)

Suppose that Sue copies the A, B, and C Analyses from the /users/joe folder and pastes them to the /users/sue folder. If the Paste Overwrite option is set to:

- **None**, then Sue keeps her A and B Analyses, and Joe's analyses are ignored. Sue gets a copy of Analysis C.
- **All**, then Sue's A and B Analyses are overwritten with Joe's, and Sue gets a copy of Analysis C.
- **Old**, then Sue keeps her A Analysis (Sue's A Analyses is not "old"), Sue's B Analysis gets overwritten by Joe's analysis (Sue's B Analysis was "old"), and Sue gets a copy of Analysis C.

18.6.2.4.2 Paste ACL The Preferences dialog contains the following options in the Paste ACL area:

- **Inherit** — Inherits the object's permissions (ACL) from its new parent folder. (Default)
- **Preserve** — Preserves the object's permissions (ACL) as it was in the original, mapping accounts as necessary.
- **Preserve Only Groups** — Same as **Preserve**, but applies to group accounts and Application Roles, not to user accounts. This is for a development to production environment in which a customer might use the same groups (such as Sales and Marketing) in both development and production. However, the users in each group might be very different, such as TestUserA and TestAdminB in development and Steve and Sue in production.
- **Create** — Preserves the object's permissions (ACL) as it was in the original, creating and mapping accounts as necessary.
- **Create Only Groups** — Same as **Create**, but applies only to group accounts and Application Roles, not to user accounts. Works in a development to production environment similarly to **Preserve Only Groups**.

Consider the following example of pasting with ACL options set. Suppose that Steve owns the /users/steve/MyFavReport folder and has permissions (ACL) "all users can read/execute, steve has full control". Joe (who has some administration privileges) logs in and copies MyFavReport, pasting it to /users/sue (which is owned by "administrator", with permissions "admins have full control, sue has full control").

If Joe sets the Paste ACL option is set to:

- **Inherit**, then the /users/sue/MyFavReport folder is owned by Joe with whatever permissions are set on the /users/sue folder (that is, "admins have full control, sue has full control").
- **Preserve**, then the /users/sue/MyFavReport folder is owned by Joe with whatever permissions were set on the /users/steve/MyFavReport folder (that is, "all users can read/execute, steve has full control"). If Joe pastes in a second Catalog Manager and if "steve" does not exist in this Catalog, then the permissions for Steve are discarded. If "steve" exists but has a different user ID, then Steve's user ID is mapped to the new one.
- **Create**, then the /users/sue/MyFavReport folder is owned by Joe with whatever permissions were set on the /users/steve/MyFavReport folder (that is, "all users can read/execute, Steve has full control"). If Joe pastes in a second Catalog Manager and if "steve" does not exist in this Catalog, then the user "steve" is created.

  This feature is used in applications whose administrators create accounts in a staging area before moving the users to the production environment.

After the paste, the object is owned by the person who performed the paste. If you have the appropriate permissions, then you can select the newly pasted object and set ownership recursively to the appropriate user.

18.6.3 Renaming Catalog Objects

You can rename objects in the catalog. This can be useful when you are migrating from a test environment to a production environment.

There are two ways to rename an object:

- Rename without reference updates — Renames the object and preserves the references to the original name that other catalog objects might have.
- Rename with reference updates — Renames the object and changes references that other objects might have to the new name (that is, original name references are not preserved). You must open the catalog in offline mode.

Caution: Do not rename the "My Dashboard" dashboards in the Users folder (displayed as _portal in Catalog Manager).

To rename an object without reference updates:

1. In Catalog Manager, open the catalog.
2. Navigate to the object to be renamed.
3. Right-click the object in the Name column and select **Rename**.
4. Type a new name for the object.

To rename an object with reference updates:

1. In Catalog Manager, open the catalog in offline mode.
2. Navigate to the object to be renamed.
3. Right-click the object in the Name column and select **Smart Rename**.
4. Type a new name for the object.

A progress bar in the lower right-hand corner of the window shows the progress of the reference updates.

18.6.4 Working with the Properties of Catalog Objects

Using the Properties option of Catalog Manager, you can:

- Create, view, edit, and delete the properties of catalog objects.
- Change attributes of catalog objects to hide them from display in Oracle Business Intelligence.

To work with the properties of a catalog object:

1. In Catalog Manager, open the catalog.
2. Navigate to the object.
3. Right-click the object in the Name column and select **Properties**.
4. Perform the necessary tasks:
   a. If you have the appropriate permissions, then select the appropriate owner for the object in the **Owner** list.

b. To change the attribute of an object, select either **Read-Only** or **Hidden**, if appropriate. A hidden object is not visible in Oracle Business Intelligence.

> **Note:** The **System** option indicates that the object is maintained internally and should not be altered.

c. To create, edit, or delete a property, use the **New**, **Edit**, or **Delete** button as appropriate.

> **Note:** The **New** button is used to create a property. You should use it only if instructed to do so by Oracle Support.

5. Click **OK**.

You can select multiple objects and update their properties or permissions simultaneously. If any of the selected objects are a folder, then you can also apply those changes recursively to all the objects in that folder's tree structure.

For example, you can set all files in the /shared/DontTouch directory to be Read-Only. Right-click the DontTouch directory and select **Properties**. In the Properties dialog, select the **Read-Only** option, select the **Apply Recursively** option, and click **OK**.

18.6.5 Setting Permissions of Catalog Objects

Permissions are used to control access to catalog objects.

To set permissions of a catalog object:

1. In Catalog Manager, open the catalog.
2. Navigate to the object.
3. Right-click the object in the Name column and select **Permissions**.

   The Permissions dialog displays these two lists:

   - **Users and groups (Explicit Permissions)** — Shows the users, groups, and application roles that have explicit permissions granted to this object.
   - **Additional users and groups** — Shows the users, groups, and application roles that have access granted through group inheritance, and users, groups, and application roles that have no access to the request.

   For details on how permissions and privileges are assigned in Presentation Services, see Section C.1.3, "How Are Permissions and Privileges Assigned to Users?"

4. If the user, group, or application role whose permissions you want to set is in the **Additional users and groups** list, then move it into the **Users and groups (Explicit Permissions)** list by selecting it and clicking the left arrow button (<).
5. (Optional) To filter the users, groups, and application roles displayed in the **Additional users and groups** list, use the **List** button with the drop-down list, and the adjacent field, as follows:

   - Enter filter criteria in the field next to the **List** button (case insensitive).

     To enter partial filter criteria, use the asterix (*) symbol. For example, enter bi* to display users or groups beginning with bi, BI, bI, and Bi.

- Select a value from the drop-down list.

  Available values are: All, User, Group, or Application Role.

6. Select the user or group in the **Users and groups (Explicit Permissions)** list.
7. Select a new permission from the list in the Permissions column, or click **Custom** from the list to display the **Custom Permissions** dialog, where you can select a combination of permissions.

   For details on what each permission means, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.
8. Select the **Apply Recursively** option to apply the changes to all the objects that the object contains.
9. Select a value from the **Replace Option** list as follows:
   - **Replace All** — Replaces the existing ACL with what is currently in the dialog.
   - **Replace Listed** — Changes only the accounts currently displayed in the dialog and leaves others unchanged.
   - **Remove Listed** — Removes only the accounts currently displayed and leaves others unchanged.
10. Click **OK**.

> **Note:** If you move a user or group from the **Users and groups (Explicit Permissions)** list to the **Additional users and groups** list, then the user or group privileges are reset to No Access. To move a user or group from one list to another, highlight it and click the right or left arrow button, as appropriate.

See Appendix C, "Managing Security for Dashboards and Analyses" for additional information on permissions and groups and users.

18.6.6 Previewing Objects from Catalog Manager

You can preview objects, such as analyses or prompts, from Catalog Manager in online mode. If you are going to preview objects from Catalog Manager, then you must identify the default browser in which to display these objects.

To set the browser preference:

1. In Catalog Manager, from the **Tools** menu, select **Preferences**.
2. In the **Select Web Browser to use for report previews** field, select the browser that is the same one that you have set to be the default browser for your operating system. You can click the **Browse** button in which you can select the executable file for the appropriate browser.
3. Click **OK**.

To preview an object:

1. In Catalog Manager, open the catalog in online mode.
2. Navigate to the object.
3. Right-click the object in the Name list and select **Preview**.

18.7 Viewing and Editing Catalog Objects in XML

You can view and edit the XML description of catalog objects such as analyses, dashboards, filters, and so on.

> **Caution:** If you change the XML code, then you change the representation of the object in the catalog. Do not edit the XML code for system catalog objects, for example in the /system/metadata directory. These objects are maintained by the system and should not be updated by users.

To view the XML description of an object:

1. In Catalog Manager, open the catalog.
2. Navigate to the object.
3. Right-click the object in the Name column and select **Properties**.
4. Click **Edit XML**.
5. When you have finished viewing the XML definition, click **Cancel**.
6. Click **OK** in the Properties dialog.

To edit the XML description of an object:

1. In Catalog Manager, open the catalog.
2. Navigate to the object.
3. Right-click the object in the Name column and select **Properties**.
4. Click **Edit XML**, then **Edit**.
5. Make the changes in the Object XML area.

> **Note:** When you edit the XML description of an object, the catalog checks only that the XML is well-formed; it does not check for any other errors.

6. Click **OK** in the Edit XML dialog.
7. Click **OK** in the Properties dialog.

18.8 Searching for and Replacing Catalog Text Using Catalog Manager

You can search for specific text in the catalog and replace it with other text using Catalog Manager. You can open the catalog in either online or offline mode.

Specifically, you can search for and replace:

- A single text string
- Multiple text strings all at once

18.8.1 Searching for and Replacing a Single Catalog Text String

Use the following procedure to search for a single text string in the catalog and replace it with other text.

To search for and replace a single text string:

1. In Catalog Manager, open the catalog in either online or offline mode.
2. From the **Tools** menu, select **XML Search and Replace**.
3. In the **Old text** field, enter the text string to search for.
4. In the **Replace with** field, enter the replacement text.
5. To make the search case insensitive, clear the **Case Sensitive** box.
6. Click **OK**.

18.8.2 About Searching for and Replacing Multiple Catalog Text Strings

You can search for and replace multiple catalog text strings all at once by importing a XML file that identifies each text string to search for and replace.

18.8.2.1 XML File Format for Searching for and Replacing Text Strings

In the search and replace XML file, you use an action element to identify each text string to search for and replace. The action elements are contained in a commands element.

The action element has the following attributes:

- command — Specifies whether you want to replace text, a column name, a subject area name, a table formula, or an entire formula. Valid values are:
  - textReplace — Replaces text other than the name of a table, column, formula, or subject area.

    The textReplace action occurs first, before other node-specific actions (for example, before renameSubjectArea).
  - renameTable — Renames the Table part of a formula alone.
  - renameColumn — Replaces the name of a column.
  - renameFormula — Renames the entire formula.
  - renameSubjectArea — Replaces the name of a subject area.
  - subjectArea — Apply this optional attribute to renameTable, renameColumn, or renameFormula.
- oldValue — Specifies the text string to search for.

  When specifying this attribute for the textReplace command, you must use the regex syntax that is described in `http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html`.

  Table 18–1 illustrates using regex syntax in search criteria.

Table 18–1 Using regex syntax in search criteria

Search String Entered	Result
a	Adds wildcards before and after your search string (for example, *a*), enabling the search to return results that contain the letter "a".
^a	Adds a wildcard after your search string (for example, a*), enabling the search to return results that begin with the letter "a".

Table 18–1 (Cont.) Using regex syntax in search criteria

Search String Entered	Result
a$	Adds a wildcard before your search string (for example, *a), enabling the search to return results that end with the character "a".
a*	Searches explicitly for strings containing a character followed by an asterix (*) for example, "a*".
?	Use a question mark (?) with a character and an asterix (*) to return zero (0) or more occurrences of a character. For example ?a* returns zero or more occurrences of the character "a".
[] or {}	Not recognized by search.

- newValue — Specifies the replacement text.
- ignoreCase — Ignores case when set to true, but becomes case sensitive when set to false. The default value is false.

18.8.2.2 Example XML File for Searching for and Replacing Text Strings

The following is an example of an XML file for searching for and replacing multiple text strings all at once:

```
<?xml version="1.0" encoding="utf-8"?>
<actions>
<action command="textReplace" oldValue="paint" newValue="HoleyShoes"
ignoreCase="true"/>
<action command="renameSubjectArea" oldValue="SnowflakeSales" newValue="GG SALES"
ignoreCase="false"/>
<action command="renameTable" subjectArea="Paint Exec" oldValue="forecast
measures" newValue="GGFCMEASURES" ignoreCase="true"/>
<action command="renameColumn" oldValue="CategoryID" newValue="GG CATID"
ignoreCase="false"/>
<action command="renameFormula" oldValue="&quot;Paint
Exec&quot;.Measures.&quot;Year Ago Dollars&quot;" newValue="&quot;Paint
Exec&quot;.Measures.&quot;GG YAGODOLLARS&quot;" ignoreCase="false"/>
<action command="renameFormula" subjectArea="&quot;Paint Exec&quot;"
oldValue="&quot;Products&quot;.Brand" newValue="GGPRODUCTS.&quot;GG BRAND&quot;"
ignoreCase="false"/>
</actions>
```

18.8.3 Searching for and Replacing Multiple Catalog Text Strings

Use the following procedure to search for and replace multiple catalog text strings all at once.

To search for and replace multiple text strings:

1. Create the XML file for searching for and replacing multiple text strings.

   For information, see Section 18.8.2, "About Searching for and Replacing Multiple Catalog Text Strings."

2. In Catalog Manager, open the catalog in offline mode.
3. From the **Tools** menu, select **XML Search and Replace**.
4. In the **Import from File** field, enter the path or click **Browse** to specify the XML file that you created in Step 1.

5. To make the search case-sensitive, select the **Case Sensitive** box.
6. Click **OK**.

18.9 Creating Reports to Display Catalog Data Using Catalog Manager

You can create reports to display catalog data for all catalog object types. You can either display the report on the screen or save it to a file.

For example, to find out which dashboards are using an analysis, you could run a Dashboard report including analyses, and search that report for the analysis. Or, to find analyses that are affected by a changed column in an RPD table, you could run an Analysis report including all columns and formulas, and then search the report for the items that must then be replaced in Catalog Manager.

When you create a report, a blank or empty field is exported as a tab character. If you are creating a report with the default of a tab as the field separator, then two tab characters in the report file indicate a blank field.

To create a report that displays catalog data:

1. In Catalog Manager, open the catalog. To create a report that shows the SQL statement that is sent to the Oracle BI Server for the object, open the catalog in online mode.
2. Select the top folder for the catalog.
3. From the **Tools** menu, select **Create Report**.
4. Select the catalog object type to create a report on.
5. To eliminate any rows that are exactly the same from the report, select the **Distinct** box.
6. Specify the columns to appear in the report in the Columns in Report list. Use the left and right arrow buttons (< and >) to move the columns between the Available Columns list and the Columns in Report list, and the plus and minus buttons (+ and -) to set the order in which columns are displayed in the report.
7. Click **OK**.
8. To save the report to a file, in the **Save report to** field, specify the path name of the file. Click the **Browse** button to display the Save As dialog for selecting the path name (if the file does not exist, then it is created).
9. Click **OK** to overwrite an existing file or click **Cancel** and enter a new name to save to a new file.
10. Select **Excel Format** to specify to create a file with a .tab extension that can be imported into Microsoft Excel 2003 or higher.
11. Click **OK**.

18.10 Archiving and Unarchiving a Catalog Using Catalog Manager

You can archive and unarchive an entire catalog or an individual catalog folder in a catalog. When you:

- Archive a catalog (that is, the catalog root folder) or an individual catalog folder, all objects in the folder and the folder's subfolders are saved in single compressed file. Properties and attributes of objects are included in the archive file.

- Unarchive a catalog or an individual catalog folder, the archive file is uncompressed and all objects in the folder and the folder's subfolders are then stored in the current offline catalog. Existing folders that have the same names as folders being unarchived are overwritten.

It is recommended for a complete catalog to archive and unarchive using 7Zip.

You also can use Oracle BI Presentation Services to archive and unarchive a catalog. For information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

To archive an entire catalog or an individual catalog folder in the catalog to a file that you specify:

1. In Catalog Manager, open the catalog in offline mode.
2. To archive:
   - The entire catalog, highlight the top-level catalog folder and from the **File** menu, select **Archive**.
   - An individual catalog folder, highlight the catalog folder and from the **File** menu, select **Archive**.
3. In the **Archive File Path** field, specify the path name of the file in which to archive the catalog or individual catalog folder. Click **Browse** to display a dialog for selecting the path name.
4. To archive the:
   - Timestamps that are assigned to the objects and folders that you are archiving, then select the **Keep file time stamps** option.

     If you do not select this option, then the archiving process does not include timestamp information. Upon unarchiving, the system applies a timestamp that indicates the time at which the object or folder is unarchived.
   - Permissions that are assigned to each object or folder, then select the **Keep permissions** option.

     If you do not select this option, then the archiving process does not include any permissions. Upon unarchiving, the system assigns the parent folder's permissions to all of the objects and folders.
5. Click **OK**.

To unarchive an entire catalog or a catalog folder:

1. In Catalog Manager, open the catalog in offline mode.
2. If you are unarchiving a catalog folder, then navigate to the location where you want to unarchive the folder.
3. From the **File** menu, select **Unarchive**.
4. In the **Archive File Path** field, specify the path name of the catalog or catalog folder to unarchive. Click **Browse** to display a dialog for selecting the path name.

   When unarchiving an entire catalog, you must select '\' (the root folder).
5. Click **OK**.

18.11 Configuring for Full-Text Catalog Search

As the administrator, you can configure a search engine to crawl and index the Oracle BI Presentation Catalog. With this configuration, you provide content designers and users with the ability to perform a full-text catalog search for objects and attachments.

This section contains the following topics:

- Section 18.11.1, "What is a Full-Text Catalog Search?"
- Section 18.11.2, "Deployment of Oracle Full-Text Catalog Search"
- Section 18.11.3, "How is Security Handled in the Full-Text Catalog Search?"
- Section 18.11.4, "Process for Configuring the Full-Text Catalog Search"

For information on content designers and users using the full-text catalog search, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

18.11.1 What is a Full-Text Catalog Search?

The full-text catalog search provides a mechanism for searching for objects in the Oracle BI Presentation Catalog that is similar to a full-text search engine. Users can search for objects by various attributes, such as name and description. Users obtain search results that correspond to the object that they have access to open. As the administrator, you must configure the use of the full-text catalog search.

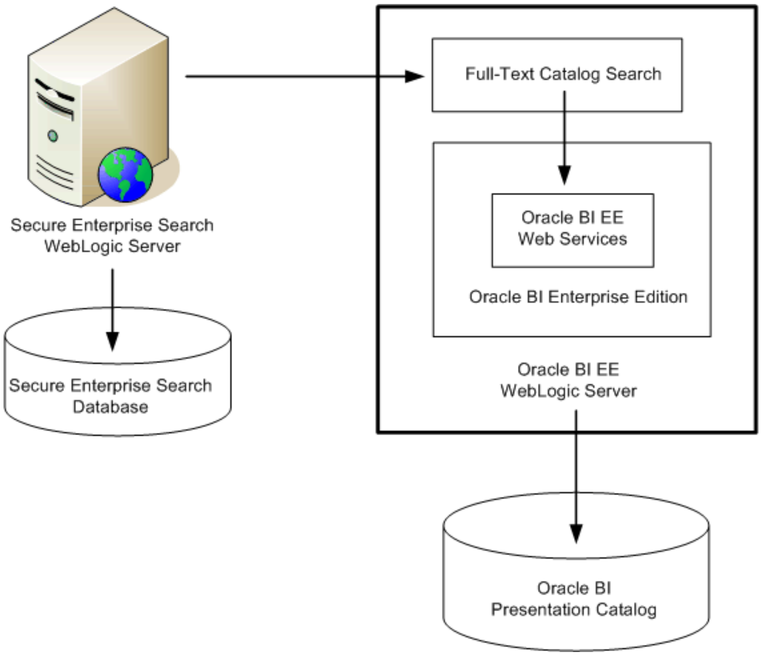
The full-text catalog search can index catalogs that contain translated text. For this release the full-text catalog search uses Secure Enterprise Search, which limits each of its registered sources to a single locale. Therefore, a multilingual translated Oracle BI Presentation Catalog can be indexed only in a single language. The user interface for Secure Enterprise Search supports multiple languages.

18.11.2 Deployment of Oracle Full-Text Catalog Search

Figure 18–1 shows a deployment diagram for the full-text catalog search. This search runs as a separate service from Oracle BI Enterprise Edition and has its own port number and URL The WebLogic Server on which Secure Enterprise Search is hosted communicates with the index for SES and with the full-text catalog search. The search uses the Web Services of Oracle BI Enterprise Edition on its WebLogic Server for accessing the Oracle BI Presentation Catalog. The services on that WebLogic Server send the requests to the catalog.

The full-text catalog search is deployed on the same WebLogic Server as Oracle BI EE. The instance for the full-text catalog search can be used with only one Oracle BI EE instance at a time. Multiple instances of Secure Enterprise Search can be connected to the same instance of the full-text catalog search.

Figure 18–1 Deployment of the Full-Text Catalog Search



18.11.3 How is Security Handled in the Full-Text Catalog Search?

The search engine enforces all the security that is defined on the Oracle BI content. Security for Oracle BI content can be defined at the object level, enforced through folder level security on the Oracle BI Server, and at the data level, enforced at run time or query time through user variables that are used to constrain the data.

In addition, the **Do Not Index** setting in the Properties dialog for an object affects whether the object can be indexed.

See *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for details on security.

18.11.4 Process for Configuring the Full-Text Catalog Search

You must configure the use of the full-text catalog search. For this release, this functionality requires Oracle Secure Enterprise Search, an application that provides crawling and indexing of the Oracle BI Presentation Catalog. The index that Secure Enterprise Search creates is used by the full-text catalog search for locating objects for which users are searching.

The following list describes the steps for configuring the full-text catalog search:

1. "Step 1: Assigning Privileges for Crawling and Indexing"
2. "Step 2: Ensuring that Secure Enterprise Search is Available"
3. "Step 3: Configuring Identity Management for Full-Text Catalog Search"
4. "Step 4: Creating the Data Source for Full-Text Catalog Search"
5. "Step 5: Editing the Schedule for Full-Text Catalog Search"
6. "Step 6: Specifying the Look and Feel of Search Results"
7. "Step 7: Enabling Incremental Crawls for Full-Text Catalog Search"
8. "Step 8: Configuring Properties for Full-Text Catalog Search Manually"

9. "Step 9: Allowing Users to Access the Full-Text Catalog Search"

18.11.4.1 Step 1: Assigning Privileges for Crawling and Indexing

The process of crawling and indexing the Oracle BI Presentation Catalog is initiated from the search engine. However, the catalog cannot be crawled and indexed unless a user with administrator privileges gives the appropriate privileges to the BISystem role.

To assign the appropriate privileges for crawling and indexing the catalog:

1. In Oracle BI Enterprise Edition, log in as an administrator.
2. Click **Administration** in the global header.
3. On the Administration tab, click **Manage Privileges**.
4. In the SOAP section, ensure that the BISystem role has the following privileges:

   Access "ReplicationService" Service
   Access "CatalogIndexingService" Service
   Access "SecurityService" Service
   Access "CatalogService" Service
   Access "XmlGenerationService" Service

For information on assigning privileges, see Section C.2.3, "Managing Presentation Services Privileges."

18.11.4.2 Step 2: Ensuring that Secure Enterprise Search is Available

The files that are required to use the full-text catalog search are copied to the appropriate directories during the installation of Oracle BI EE. Users cannot access the full-text catalog search if a supported search engine is not installed and configured for use with the Oracle BI Presentation Catalog.

For this release of Oracle BI EE, Oracle Secure Enterprise Search (SES) is the supported search engine. You can use SES Release 11.1.2. You can install SES either before or after you install Oracle BI EE and on either the same computer or a different one. For information on SES, display the following page from Oracle Technology Network:

`http://www.oracle.com/technology/products/oses/index.html`

Before proceeding, ensure that Secure Enterprise Search is installed at your site and that you know the URL to the Secure Enterprise Search administration tool. You use the administration tool to configure crawling and indexing of the Oracle BI Presentation Catalog. Use a URL in the following format to access the administration tool:

http://*host*:*port*/search/admin/control/login.jsp

where *host* is the computer on which Secure Enterprise Search is installed and *port* is the port for the administration tool.

18.11.4.3 Step 3: Configuring Identity Management for Full-Text Catalog Search

Use the Secure Enterprise Search administration tool to configure identity management for the full-text catalog search.

To configure Identity Management for the full-text catalog search:

1. Access the Secure Enterprise Search administration tool.
2. Display the Identity Management Setup page using steps such as the following:

Select **Global Settings**, then **Identity Management Setup** in the System section.

3. In the list of available identity plug-ins, select **Oracle Fusion**.
4. On the Activate Identity Plug-in tab, specify the following:
   - HTTP endpoint for authentication: The URL for the full-text catalog search security service, in the following format:

     http://*host*:*port*/bisearch/crawler/SecurityService

     where *host* is the computer on which the full-text catalog search service is installed and *port* is its port number.
   - User ID: The ID of a valid Oracle Business Intelligence user.
   - Password: The password of the user.
5. Click **Finish**.

18.11.4.4 Step 4: Creating the Data Source for Full-Text Catalog Search

Use the Secure Enterprise Search administration tool to configure the Oracle BI Presentation Catalog as a source that can be crawled and indexed.

To configure the data source for the full-text catalog search:

1. Access the Secure Enterprise Search administration tool.
2. Select the Home tab, then the Sources tab.
3. In the Source Type list, select **Oracle Fusion**, then click **Create**.
4. On the Source Configuration tab, specify the following:
   - Name: Enter a name for the data source.
   - Configuration URL: The URL for the configuration file for the full-text catalog search, in the following format:

     http://*host*:*port*/bisearch/crawler/oracle.biee.search.BISearchableTreeObject/ConfigFeed?forceInitialCrawl=true

     where *host* is the computer on which the full-text catalog search service is installed and *port* is its port number.
   - Authentication Type: The type of authentication for users. Select the default value of **Native**, which allows the full-text catalog search to handle authentication.
   - User ID: The ID of a valid Oracle Business Intelligence user.
   - Password: The password of the user.
   - Realm: Leave this blank.
   - Scratch Directory: Specify a temporary directory where Secure Enterprise Search can store temporary files.
5. On the Authorization tab, specify the following:
   - Authorization: Select **ACLs Controlled by the Source**.
   - HTTP endpoint for authorization: The URL for the full-text catalog search security service, in the following format:

     http://*host*:*port*/bisearch/crawler/SecurityService where *host* is the computer on which the full-text catalog search service is installed and *port* is its port number.

- User ID: The ID of a valid Oracle Business Intelligence user.
- Password: The password of the user.
- Business Component: Enter the following value: oracle.biee.search.BISearchableTreeObject
- Display URL Prefix: The URL for the search functionality in Oracle BI Enterprise Edition, in the following format:

  http://*host*:*port*/bisearch/urlbuilder

  where *host* is the computer on which Oracle BI Enterprise Edition is installed and *port* is its port number.

6. Click **Finish**.

18.11.4.5 Step 5: Editing the Schedule for Full-Text Catalog Search

When you create a data source in the Secure Enterprise Search administration tool for the Oracle BI Presentation Catalog, a schedule for crawling and indexing of that source is created automatically. You can use the administration tool to edit the schedule.

There are two types of crawls:

- Full crawl — Crawls the entire catalog. Use this type of crawl the first time that a catalog is crawled or when a large number of changes have occurred since the previous crawl. You should generally schedule a full crawl when most users are not accessing the system, such as over the weekend or a holiday. You must perform a full crawl with replication enabled on a catalog before you can perform an incremental crawl.
- Incremental crawl — Crawls or actually indexes only those objects in the catalog that have changed. A changed object is one that has been created, edited, or deleted. An incremental crawl does still crawl and build the entire catalog tree structure, but only the changed files are re-indexed. This provides a quick way to update the search index without re-indexing the entire catalog tree.

  Incremental crawls occur only if you enable replication, as described in "Step 7: Enabling Incremental Crawls for Full-Text Catalog Search".

After the catalog is crawled fully or incrementally, the objects are indexed so that users can easily search for them as part of the full-text catalog search.

By default, all objects in the catalog are set to be indexed during a crawl. Content designers can use the Properties dialog for an object to set the Do Not Index property of that object. Setting this property prohibits the object from being crawled and indexed. For more information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

18.11.4.6 Step 6: Specifying the Look and Feel of Search Results

You can specify the look and feel of the results that are returned by the full-text catalog search by using two files that are named bisearch.xslt and bisearch.css. These files are located in the following directory:

ORACLE_HOME\user_projects\domains\bifoundation_domain\config\fmwconfig\biinstances\coreapplication

Use the Secure Enterprise Search administration tool to configure the look and feel for the full-text catalog search.

To configure the look and feel for the full-text catalog search:

1. Access the Secure Enterprise Search administration tool.
2. Select **Global Settings**, then **Configure Search Result List** in the Out-of-Box Query Application section.
3. On the Configure Search Result List tab, specify the following:
   - Select **Use Advanced Configuration**.
   - Select all the attributes in the Not Included list that start with "BI" and move them to the Included list.
   - In the field that is labeled "Enter an XSLT to convert XML result data into HTML," copy and paste the contents of the bisearch.xslt file.
   - In the field that is labeled "Enter a CSS to style the HTML," copy and paste the contents of the bisearch.css file.

18.11.4.7 Step 7: Enabling Incremental Crawls for Full-Text Catalog Search

In order for incremental crawls to occur, you must first do the following:

1. Enable the replication of the Oracle BI Presentation Catalog.
2. Perform a full crawl on the catalog.

To enable replication, set the Replication element to true in the instanceconfig.xml file. The following procedure provides information on configuring to enable replication.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the setting for replication:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the Replication element.

   The Replication element is set to false by default. If you do not set it to true, then only full crawls of the catalog occur, no matter how minor the changes to the catalog since the last full crawl.
3. Include the element and its ancestor elements as appropriate, as shown in the following example:

```
<ps:Catalog>
   <ps:Replication>
       <ps:Enabled>true</ps:Enabled>
   </ps:Replication>
</ps:Catalog>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

After you have enabled replication, use a URL in the following format to start an incremental crawl. Include false at the end of the URL to perform an incremental crawl, and include true at the end to perform a full crawl.

http://*host:port*/bisearch/crawler/oracle.biee.search.BISearchableTreeObject/ConfigFeed?forceInitialCrawl=false

18.11.4.8 Step 8: Configuring Properties for Full-Text Catalog Search Manually

You can configure properties for the full-text catalog search by manually editing the BISearchConfig.properties file. Because the properties for the full-text catalog search include specifications for the URLs of Web Services that are used in the full-text catalog search, you can set these properties, if URLs change.

Table 18–2 describes the configuration properties for the full-text catalog search, which reside in the BISearchConfig.properties file. The BISearchConfig.properties file is installed into the *ORACLE_HOME*\user_projects\domains\bifoundation_domain/config/fmwconfig/biinstances/coreapplication directory on the WebLogic Server for Oracle BI Enterprise Edition.

Table 18–2 Configuration Properties for the Full-Text Catalog Search

Name	Description	Default Value
BIServerURL	Specifies the URL for the Oracle BI Server.	NA
BISearchServerURL	Specifies the URL for the servlet for the full-text catalog search.	NA
CrawlingUserKey	Specifies the key for the credentials that are used to crawl the catalog. You should not edit this value.	NA
ExcludePaths	Specifies the list of directories from the catalog that should *not* be crawled. Delimit directories with a semicolon (;).	"/system/*;/shared/*"
IncludePaths	Specifies the list of directories from the catalog that should be crawled. Delimit directories with a semicolon (;).	"*"
RootNode	Specifies the node from which all crawling originates.	"/"

18.11.4.9 Step 9: Allowing Users to Access the Full-Text Catalog Search

When you have configured Secure Enterprise Manager to crawl and index the Oracle BI Presentation Catalog, you can notify content designers and users that the Search page is available for performing the full-text catalog search.

18.11.4.9.1 Providing Designers with the URL You can provide designers with the URL to display the page for the full-text catalog search. Designers can bookmark this page and use it with actions or as a shortcut on a dashboard page.

The person who installs Secure Enterprise Search sees a page that provides the URL in the following form:

http://*host*:*port*/search/query/search

where *host* is the computer on which Secure Enterprise Search is installed and *port* is the port for the administration tool. The default port is 7777.

18.11.4.9.2 Making the Shortcut Available from the Oracle BI EE Home Page You can configure the Oracle BI EE Home page to include a shortcut in the All Content area that allows users to display the page for the full-text catalog search. To do so, edit the instanceconfig.xml file to include the SESSearchURL element, as described in the following procedure.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually add the shortcut for the full-text catalog search to the Home page:

1. Open the instanceconfig.xml file for editing, as described in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings."
2. Locate the section in which you must add the SESSearchURL element.

   The value for the SESSearchURL element is the URL for the Secure Enterprise Manager page.
3. Include the element and its ancestor elements as appropriate, as shown in the following example:

```
<Server>
<ps:SESSearchURL
xmlns:ps="oracle.bi.presentation.services/config/v1.1">http://myhost:7777/searc
h/query/search</ps:SESSearchURL>
</Server>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

18.12 Replicating Oracle BI Presentation Catalogs

You can copy and merge the contents of selected catalog folders between instances of Presentation Services. You can split the replication configuration into individual tasks. The replication task is an instruction to periodically merge catalog contents for specified catalog folders from one server to another. Two-way replications are possible.

A typical scenario where replication is useful is when you use one instance of Presentation Services to prepare and publish shared analyses, and you use two or more production instances to support the users. In this scenario, you share the catalog from Instance 1 to two production instances and share the catalogs on each production instance with the other. To accomplish this, you configure the following replication tasks in the configuration file for the Presentation Services Replication Agent:

- /shared: Instance1 to Instance2
- /shared: Instance1 to Instance3
- /users: Instance2 to Instance3
- /users: Instance3 to Instance2

When an instance of Presentation Services participates in replication, either as a publisher or as a subscriber, it tracks changes made to catalog objects that are marked to be replicated and keeps them internally in replication log files. Another instance of Presentation Services can make a SOAP call to request to export those changes to a file or to import and replay modifications recorded in a file exported earlier from another instance. The Presentation Services Replication Agent uses SOAP to manage replication-related activities on all instances of Presentation Services and performs import and export operations. For information on these SOAP calls, see *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition*.

18.12.1 About Replication Log Files

Replication log files record modifications made to catalog objects that are subject to replication. There are two types of replication log files:

- Change log files — Contain records of modifications to catalog objects that were made locally. These files are stored in *presentationcatalogpath*/replication/changelog folder, where *presentationcatalogpath* is the full path to the catalog folder.

- Playback log files — Contain records of modifications to catalog objects that were replayed from other instances of Presentation Services. These files are stored in *presentationcatalogpath*/replication/playback folder.

Replication log files are stored on disk indefinitely, until you periodically remove them. When performing replication operations, Presentation Services reads only replication log files from the time interval that is specified by the ReadLogRecordsSinceHoursAgo element in the instanceconfig.xml file. For information on this element, see Section 18.12.5, "Editing the instanceconfig.xml File for Replication."

While you might want to review replication log files, for example, for troubleshooting purposes, you should never edit log files manually.

18.12.1.1 Replication Log File Records

A replication log file contains different types of records, with each record consisting of several fields that are separated by commas. The most important types of records in a replication log file are described in Table 18–3.

Table 18–3 Description of Records in Replication Log Files

Type	Written	Contents
File header record	At the beginning of each log file	Size, Type, Timestamp, Version, Flags
Startup record	Each time Presentation Services starts	Size, Type, Timestamp
Before change record	When an Oracle BI Presentation Catalog object is about to be changed	Size, Type, Timestamp, Change sequence number, Change type
After change record	Immediately after a catalog operation has been completed	Size, Type, Timestamp, Change sequence number, Operation outcome
File finished record	As the last record in a replication log file	Size, Type, Timestamp, Name of next log file

The following list describes some fields in a replication log file:

- **Size.** The size of the record in bytes and encoded as a hexadecimal number.
- **Type.** The type of record, such as the following:
  - **H.** File header record
  - **S.** Presentation Services startup record
  - **B.** Before change record
  - **A.** After change Record
  - **F.** File finished record
- **Timestamp.** The timestamp of the record, written as a hexadecimal number that represents the number of seconds from midnight (00:00:00), January 1, 1970 UTC.
- **Change sequence number.** A sequence number that should be the same in the corresponding Before change and After change records.

- **Change type.** The type of change, written as a hexadecimal number that represents a combination of the following flags:
  - **1.** Insert
  - **2.** Delete
  - **4.** Write
  - **8.** Change catalog object information
  - **16.** Change owner
  - **32.** Change security
- **Operation outcome.** The operation result, which is one of the following:
  - **0.** Fail
  - **1.** Success

18.12.2 Configuring to Replicate an Oracle BI Presentation Catalog

To enable replication of an Oracle BI Presentation Catalog, perform the steps that are described in the following sections:

- Section 18.12.2.1, "Copying Oracle BI Presentation Catalog Files"
- Section 18.12.3, "Using the Oracle BI Presentation Services Replication Agent"
- Section 18.12.4, "Creating the config.xml File for Replication"
- Section 18.12.5, "Editing the instanceconfig.xml File for Replication"

18.12.2.1 Copying Oracle BI Presentation Catalog Files

The Presentation Services Replication Agent copies changes made to an Oracle BI Presentation Catalog. Because it copies only changes and not the entire contents, you must first make a copy of the source catalog on each of the subscribing instances of Presentation Services.

18.12.3 Using the Oracle BI Presentation Services Replication Agent

You can run the Replication Agent on any computer that has network access to both instances of Presentation Services. This agent handles all the replication tasks and provides command line options.

The Replication Agent (whose file name is sawrepaj) is a utility that performs common replication tasks such as copying, exporting, importing, and marking for replication.

The Replication Agent requires information about the instances of Presentation Services and the replication tasks. This information is stored in the config.xml file, and described in Section 18.12.4, "Creating the config.xml File for Replication."

The command line for the Replication Agent uses the following format:

- In UNIX:

  ```
  sawrepaj.sh [/C path] command [command parameters]
  ```

  The sawrepaj.sh file is located in *ORACLE_HOME*\bifoundation\web\bin\sawrepaj.

- In Windows

  ```
  sawrepaj.bat [/C path] command [command parameters]
  ```

The sawrepaj.bat file is located in *ORACLE_HOME*\bifoundation\web\bin\sawrepaj.

The path following the "/C" specifies the path to the config.xml file. The "command" options are described in the following sections:

- Section 18.12.3.1, "mark"
- Section 18.12.3.2, "remotecopy"
- Section 18.12.3.3, "run"

18.12.3.1 mark

Adds or removes the specified catalog folders from the list of catalog folders that are to be replicated on all or the specified computers that run Presentation Services. The list of catalog folders to be replicated is stored in a configuration item that is located in the *presentationcatalogpath*/root/system/replication directory, where *presentationcatalogpath* is the full path to the Oracle BI Presentation Catalog directory.

Adding a catalog folder to the list does not mean that the corresponding catalog object is immediately replicated. Only future modifications are noted in the replication log file and subsequently replicated.

When you replicate and the Oracle BI Presentation Catalog exists on another instance, the list of catalog folders to be replicated might be incorrect. Use the mark command to maintain the list of catalog folders to be replicated.

The syntax for the mark command is:

```
sawrepaj [/C path] mark {all|servername} [/n] [catalogfolders]
```

- **all | servername.** The name of the computer that is running Presentation Services or "all," on which to run the mark command. If you use a specific server, then the name that you specify must match the name attribute of the corresponding server element in the config.xml file.
- **/n**. If present, indicates that the catalog folder should be removed from the list of catalog folders to be replicated. Otherwise, the folder is added.
- **catalogfolders.** A list of catalog folders to add or remove from the list of catalog folders to be replicated. Separate each folder name in the list with a space. To add or remove the entire Oracle BI Presentation Catalog (including all folders and such information as system privileges, Catalog Group membership, and so on), use /.

  If you do not specify any folders, then the mark command is executed on every folder for the servers that are specified in the config.xml file.

The following line shows a sample command for mark:

```
sawrepaj /C MyOracleHome\bifoundation\web\bin\sawrepaj mark all /
```

18.12.3.2 remotecopy

Exports the contents of the specified catalog folders from the source computer and imports them into the folders on the destination computer.

The syntax for the remotecopy command is:

```
sawrepaj [/C path] remotecopy sourceServer destinationServer catalogfolders
```

- **sourceServer.** The name of the source computer as specified in the config.xml file.

- **destinationServer.** The name of the destination computer as specified in the config.xml file.
- **catalogfolders.** A list of catalog folders to copy remotely. Separate each folder name in the list with a space. If you do not specify any folders, then the remotecopy command is executed on every folder for the servers that are specified in the config.xml file.

The following line shows a sample command for remotecopy:

```
sawrepaj /C MyOracleHome\bifoundation\web\bin\sawrepaj remotecopy Server1 Server2
users shared
```

18.12.3.3 run

Executes all replication tasks that are specified in config.xml file and that have not expired. This command has no run-time parameters.

The syntax for the run command is:

```
sawrepaj [/C path] run
```

The following line shows a sample command for run:

```
sawrepaj /C MyOracleHome\bifoundation\web\bin\sawrepaj run
```

18.12.4 Creating the config.xml File for Replication

You must create the Replication Agent's configuration file to specify which folders on which instances of Presentation Services are to be replicated. The config.xml file is located in the *ORACLE_HOME*\bifoundation\web\bin\sawrepaj directory.

The following sections describe the config.xml file:

- Section 18.12.4.1, "Structure of the Configuration File"
- Section 18.12.4.2, "Elements of the Configuration File"
- Section 18.12.4.3, "Example config.xml File"

For information on manual editing, see Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings."

18.12.4.1 Structure of the Configuration File

The structure of the Replication Agent's config.xml file is as follows:

```
<Config>
   <General>
      <ExportDirectory/>
      <LogExpiresHours/>
      <RetryAttempts/>
   </General>
   <Server>
      <Property/>
   </Server>
   <Folderset>
      <Folder/>
   </Folderset>
   <ReplicationTask/>
</Config>
```

18.12.4.2 Elements of the Configuration File

Table 18–4 describes the elements that you can set in the Replication Agent's config.xml file. You must enter the elements manually as they do not exist by default.

Table 18–4 Replication Agent Configuration Elements in config.xml

Element	Description
Config	Serves as the root element for the configuration file.
Config/General	Contains general settings that are applicable to all instances and replication tasks.
General/ExportDirectory	Contains the UNC path to the shared directory where export files should be placed. Must be accessible by the same name from all instances of Presentation Services. In addition, the user account under which an instance of Presentation Services is running must have read/write permissions to this directory.
General/LogExpiresHours	Specifies the time in hours (as a double value) that replication logs expire on all instances of Presentation Services. Set this element to the value of the ReadLogRecordsSinceHoursAgo element that is the lowest among all instances of Presentation Services that are participating in replication. For information on the ReadLogRecordsSinceHoursAgo element, see Section 18.12.5, "Editing the instanceconfig.xml File for Replication."
General/RetryAttempts	Specifies the number of times an attempt to change an object is to be retried. (An attempt to change an object might be retried, for example, if an object was locked.)
Config/Server	Defines connection information for each computer for Presentation Services. Its attributes are: ■ **name.** The logical ID of the server, which is used to identify it in replication tasks and in the sawrepaj command line. ■ **user.** The user name. ■ **pwd.** The password.

Table 18–4 (Cont.) Replication Agent Configuration Elements in config.xml

Element	Description
Config/Server/Property	Defines connection parameters for the Presentation Services (SAWBridge) server, and the transport protocol, its attributes are: Defines server name and value: **Note:** The settings that you use here are determined by whether the value for the transport protocol below is set to use HTTP (N) or socket based TCP (Y). ■ **name** - Defines the server name, for example: oracle.bi.presentation.url - (when using HTTP protocol) oracle.bi.presentation.sawservers - (when using socket based TCP protocol). ■ **value** - Defines the server address, for example: http://localhost/analytics/saw.dll - (when using HTTP protocol) the mapping alias of SAWBridge servlet localhost:9710 - (when using socket based TCP protocol). Defines the transport protocol name and value: **Note:** The value that you use here determines the name and value that you use above. ■ **name** - The transport protocol name, for example: oracle.bi.presentation.usesoaptcptransport ■ **value** - The transport protocol for connecting to the Presentation Server can be either of the following: **N** - Use HTTP protocol (Default value). **Y** - Use socket based TCP protocol.
Config/Folderset	Defines a list of catalog folders. Its only attribute is name, which you can set to the logical ID of the server or to "all".
Folderset/Folder	Adds a folder to the folderset. Specifies the full path to the catalog folder.
Config/ReplicationTask	Defines the replication tasks. Its attributes are: ■ **source.** The name of the source computer. This must match the name of an instance that is defined in the Server element. ■ **destination.** The name of the destination computer. This must match the name of an instance that is defined in the Server element. ■ **folders.** The name of the folderset. This must match a folderset that is defined in a Folderset element. ■ **lastPerformed.** The timestamp of the last successful run of this task. The sawrepaj utility updates this value. ■ **localChanges.** Specifies whether changes made directly on the source computer are to be exported: **true.** Changes are to be exported. (Default) **false.** Changes are not to be exported. ■ **remoteChanges.** Specifies whether changes that were made on another server and that were replicated on the source server are to be reexported: **true.** Changes are to be re-exported. **false.** Changes are not to be re-exported. (Default)

18.12.4.3 Example config.xml File

The following is an example of the config.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Oracle BI Presentation Services Replication Agent Configuration File -->
<!-- replicate entire catalog from host1 to host2 -->

<Config>
      <General>
         <ExportDirectory>\\host1\shared</ExportDirectory>
         <LogExpiresHours>48</LogExpiresHours>
      </General>
      <Server name="1" user="<admin1user>" pwd="<admin1password>">
        <Property name= "oracle.bi.presentation.url"
        value ="http://host1.com:<port>/analytics/saw.dll"/>
        <Property name="oracle.bi.presentation.usesoaptcptransport" value="N"/>
      </Server>

      <Server name="2" user="admin2user" pwd="admin2password">
      <Property name="oracle.bi.presentation.sawservers"
        value ="localhost:10712"/>
      <Property name="oracle.bi.presentation.usesoaptcptransport" value="Y"/>
      </Server>

      <Folderset name="all">
         <Folder>/</Folder>
      </Folderset>

      <ReplicationTask destination="2" source="1" folders="all" />
   </Config>
```

18.12.5 Editing the instanceconfig.xml File for Replication

For each instance of Presentation Services that contains a catalog to replicate, you must edit the instanceconfig.xml file for the following:

- Enable replication. By default, the replication functionality is disabled for instances of Presentation Services.
- Configure replication for a clustered environment. In a clustered environment, several instances of Presentation Services can share the same Oracle BI Presentation Catalog. For replication to work properly, you must:
  - Ensure that each instance of Presentation Services writes to its own log file to avoid concurrency issues.
  - Ensure that changes made by all instances are taken into account during import and export operations.
- Specify the number of records to be kept in a replication log file.
- Specify the time period (in hours) from which replication log files are to be read for replication operations.

The following procedure provides information on configuration changes that you can make for replication. Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings for the replication of catalogs:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the elements that are described in Table 18–5.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <CatalogReplication>
    <Enabled>False</Enabled>
    <ReadLogRecordsSinceHoursAgo>120</ReadLogRecordsSinceHoursAgo>
    <RecordsInFileLimit>4000</RecordsInFileLimit>
  </CatalogReplication>
  <CatalogReplicationCluster>
    <IDsinCluster>ID03</IDsinCluster>
    <MyInstanceID>%INSTANCEID%</MyInstanceID>
  </CatalogReplicationCluster>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Presentation Services.

Table 18–5 Elements for Configuring Catalog Replication

Element	Description	Default Value
Enabled	Specifies whether the instance of Presentation Services is to participate in replication, as either a publisher or subscriber.	False
ReadLogRecordsSinceHoursAgo	Specifies the time period (in hours) from which replication log files are to be read for replication operations.	168
RecordsInFileLimit	Specifies the number of records to be kept in a replication log file.	5000
IDsInCluster	Identifies, by ID, all instances of Presentation Services that are in the cluster. Separate each ID by a comma. This element is required in a clustered environment.	NA
MyInstanceID	Identifies the suffix (%INSTANCEID%) to be used for the log files generated by this instance of Presentation Services. Log files are named as follows: ■ For change log files: sawchange_%INSTANCEID%_%SEQNUM%.log ■ For playback log files: sawplayback_%INSTANCEID%_%SEQNUM%.log This element is required in a clustered environment.	NA

18.12.6 Resuming Replication If Replication Log Files Are Unavailable

If, for some reason, replication log files are unavailable in an instance of Presentation Services, then you can resume replication by copying over the Oracle BI Presentation Catalog folders that the instance subscribes to. However, this method loses the catalog contents that were not replicated to other instances. The following method preserves all catalog contents.

> **Note:** If you must resume replication after Presentation Services has been offline for a time period in excess of that specified in the ReadLogRecordsSinceHoursAgo element in the instanceconfig.xml file, then you can adjust the value of ReadLogRecordsSinceHoursAgo so that log files in the offline period are replicated. Then, reset ReadLogRecordsSinceHoursAgo back to its original setting.

To resume replication if replication log files are unavailable:

1. Use Catalog Manager to erase all replication logs in the *presentationcatalogpath*/replication/changelog and *presentationcatalogpath*/replication/playback folders, where *presentationcatalogpath* is the full path to the Oracle BI Presentation Catalog folder.

   For information about Catalog Manager, see Chapter 11, "Configuring Connections to External Systems."

2. Set the UseReplication element in the instanceconfig.xml file to false.

   For information, see Section 18.12.5, "Editing the instanceconfig.xml File for Replication."

3. Start the instance of Presentation Services.

4. Delete or rename folders that are replicated from other instances and copy them over from other instances using the sawrepaj remotecopy command.

5. Set the UseReplication element in instanceconfig.xml to true.

6. Restart the instance of Presentation Services.

7. Restore the list of folders to be replicated using the sawrepaj mark command on the instance of Presentation Services.

8. Re-enable replication tasks by editing the configuration file to delete the lastPerformed attribute from all ReplicationTask elements that have the instance of Presentation Services as an import or export target.

Part VI

Advanced Configuration Settings

Part V describes configuration settings that are required for deploying a system. This part describes advanced configuration settings that are not required but are optional and advanced settings for fine-tuning a deployment.

This part contains the following chapters:

19 Configuring and Managing Analyses and Dashboards

This chapter describes how to configure and manage analyses and dashboards and the objects that they contain, such as views. For information on how content designers work with analyses and dashboards, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

End users with appropriate privileges can modify personal and shared dashboards, including the addition of pages and content. End users cannot create analyses and dashboards.

This chapter contains the following sections:

- Section 19.1, "Managing Dashboards"
- Section 19.2, "Performing General Configuration Tasks for Analyses"
- Section 19.3, "Configuring for Displaying and Processing Data in Views"
- Section 19.4, "Manually Changing Presentation Settings"
- Section 19.5, "Controlling Access to Saved Customization Options in Dashboards"
- Section 19.6, "Blocking Analyses in Answers"
- Section 19.7, "Specifying View Defaults for Analyses and Dashboards"
- Section 19.8, "Configuring for Write Back in Analyses and Dashboards"

19.1 Managing Dashboards

Before you create shared dashboards, ensure that you have planned the Oracle BI Presentation Catalog directory or folder structure and security strategy. In general, to create a shared dashboard, you first create the dashboard and add content using the Dashboard Builder. You can also assign permissions to access the dashboard. Users who are members of multiple application roles or Catalog groups can select the dashboard that they display by default from all of the dashboards to which they have permissions.

The following list provides other resources with information on dashboards:

- Guidelines for creating a shared dashboard, within the broader context of the Oracle BI Presentation Catalog structure and security framework, are provided in Section C.4, "Providing Shared Dashboards for Users."
- Information about shared folder structures in the Oracle BI Presentation Catalog is provided in Chapter 18, "Configuring and Managing the Oracle BI Presentation Catalog."

- Information about permissions is provided in Section C.1.3, "How Are Permissions and Privileges Assigned to Users?"
- Details for authorizing users to act for others, which allows them to access the other users' dashboards, is provided in Section C.5, "Enabling Users to Act for Others."

19.2 Performing General Configuration Tasks for Analyses

This section describes general tasks that you can perform to configure for the creation of analyses. It includes the following sections:

- Section 19.2.1, "Providing Access to Metadata Dictionary Information"
- Section 19.2.2, "Supporting Nested Folders, Navigation, and Drill Down"

19.2.1 Providing Access to Metadata Dictionary Information

When creating analyses, content designers might need more information about subject areas, folders, columns, or levels (such as relationships to other metadata objects) to guide them. You can provide content designers with this information by allowing them access to the metadata dictionary for the repository. The metadata dictionary describes the metrics that are contained within the repository and the attributes of repository objects. The metadata dictionary output is a static set of XML documents.

To provide access to metadata dictionary information:

1. Ensure that the metadata dictionary has been generated and the files have been stored in an appropriate location. For information about generating the metadata dictionary files, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.
2. Set the DictionaryURLPrefix element within the ServerInstance element in the instanceconfig.xml file to one of the following values. The value that you specify depends on the Web servers in use.

   - The prefix for the name of the directory in which you have stored the XML files. The directory must have been specified as a shared directory for the Web server, and the Web server must be the same one that is used by Oracle BI EE.

     For example, suppose that you stored the XML files for the metadata dictionary in a directory called demo1 under the metadictionary directory. Suppose that the metadictionary directory is specified as a shared directory for the Web server, which is also used by Oracle BI EE. Then you specify the following value for the DictionaryURLPrefix element:

     ```
     <DictionaryURLPrefix>demo1/</DictionaryURLPrefix>
     ```

     See the documentation for your Web server for information on sharing directories.

   - The URL that points to the directory in which you have stored the XML files. Use a value such as this when the files for the metadata dictionary are stored in the directory structure for a Web server that is not being used by Oracle BI EE. For example:

     ```
     <DictionaryURLPrefix>http://10.10.10.10/metadictionary/dem
     o1</DictionaryURLPrefix>
     ```

   The following shows an example setting in the instanceconfig.xml file:

   ```
   <WebConfig>
   ```

```
  <ServerInstance>
    <SubjectAreaMetadata>
      <DictionaryURLPrefix>demo1</DictionaryURLPrefix>
    </SubjectAreaMetadata>
  </ServerInstance>
</WebConfig>
```

For information about working in the Oracle BI Presentation Services configuration file (instanceconfig.xml), see Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings."

3. Grant the Access to Metadata Dictionary privilege to the appropriate content designers. For information about privileges, see Section C.2.3.3, "Default Oracle BI Presentation Services Privilege Assignments."

For details on how content designers can view metadata dictionary information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

19.2.2 Supporting Nested Folders, Navigation, and Drill Down

The Oracle BI Administrator can set up subject areas in ways that assist content designers who work with analyses. *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* provides complete information on setting up subject areas. The following list describes the features of subject areas that assist content designers:

- To make selections easy for content designers to discern in the Subject Areas pane when creating analyses, the administrator can set up the Presentation layer in the Oracle BI Administration Tool to give the appearance of nested folders. For example, the administrator can make the Sales Facts folder appear as a subfolder in the Facts folder.

- When content designers create analyses, they can allow users to go to related analyses and content. If the Oracle BI Administrator sets up dimensions and dimensional hierarchies for the subject area, then users can drill down on data results that are presented in graphs, tables, and pivot tables to obtain more detailed information.

  There are no specific privilege settings that control access to navigation and drill down features, which are available to all users.

19.3 Configuring for Displaying and Processing Data in Views

You can configure various options that change the display and processing of data in views. See also Section 7.3.3, "Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables" and Section 7.3.4, "Using Fusion Middleware Control to Set the Maximum Number of Rows Processed to Render a Table" for related information.

This section contains the following topics:

- Section 19.3.1, "Manually Configuring for Data in Views"
- Section 19.3.2, "Manually Configuring for Graphs and Gauges"
- Section 19.3.3, "Manually Changing Alternating Bar Color"

19.3.1 Manually Configuring for Data in Views

You can configure various options that change the processing and display of data in views, as described in the following sections:

- Section 19.3.1.1, "Manually Configuring Cube Settings for Pivot Tables and Graphs"
- Section 19.3.1.2, "Manually Configuring Settings for Data in Views"

19.3.1.1 Manually Configuring Cube Settings for Pivot Tables and Graphs

You can use settings within the Cube element to affect the display and processing of data in pivot tables and graphs. The settings also take effect for XMLA export.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the Cube settings:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the Cube section, in which you must add the following elements:
   - CubeMaxRecords — Specifies the maximum number of records that are returned by an analysis for the view to process. This roughly governs the maximum number of cells that can be populated in a view; unpopulated cells in a sparse view do not count. The default is 40000.
   - CubeMaxPopulatedCells — Specifies the maximum number of cells in a view that can be populated with data from the Oracle BI Server. The default is 120000.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <Views>
    <Cube>
      <CubeMaxRecords>30000</CubeMaxRecords>
      <CubeMaxPopulatedCells>120000</CubeMaxPopulatedCells>
    </Cube>
  </Views>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

19.3.1.2 Manually Configuring Settings for Data in Views

You can configure a similar group of settings that affects the display of data in tables, pivot tables, and graphs. While the settings are mainly the same, you must include the element within each appropriate parent element to override the default setting that applies to that view. For example, the views all use the MaxVisiblePages element. You must include that element within each of the Table, Pivot, and Charts parent elements, to override the default value of that setting for each of those view types.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings that change the display of data in views:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the Table, Pivot, and Charts parent sections, in which you must add the elements that are described in Table 19–1.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example.

```
<ServerInstance>
  <Views>
      <Table>
        <MaxVisiblePages>1000</MaxVisiblePages>
        <MaxVisibleRows>500</MaxVisibleRows>
        <MaxVisibleSections>25</MaxVisibleSections>
        <DefaultRowsDisplayed>30</DefaultRowsDisplayed>
      </Table>
      <Pivot>
        <MaxVisibleColumns>300</MaxVisibleColumns>
        <MaxVisiblePages>1000</MaxVisiblePages>
        <MaxVisibleRows>500</MaxVisibleRows>
        <MaxVisibleSections>25</MaxVisibleSections>
        <DefaultRowsDisplayed>30</DefaultRowsDisplayed>
      </Pivot>
      <Charts>
        <MaxVisibleColumns>2000</MaxVisibleColumns>
        <MaxVisiblePages>1000</MaxVisiblePages>
        <MaxVisibleRows>2000</MaxVisibleRows>
        <MaxVisibleSections>25</MaxVisibleSections>
        <JavaHostReadLimitInKB>4096</JavaHostReadLimitInKB>
      </Charts>
  </Views>
</ServerInstance>
```

   Note that this example does not include parameters that might exist in the file, but that are centrally managed by Fusion Middleware Control and cannot be changed manually.

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

Table 19–1 describes the common elements that affect the display of data in views. If the user exceeds these values, then the Oracle BI Server returns an error message when the view is rendered.

Table 19–1 Common Elements for Manually Changing the Display of Data in Views

Element	Description	Default Value	Applicable Views
DefaultRowsDisplayed	Specifies the default number of rows to display in the view.	30	Pivot Table, Table
MaxVisibleColumns	Specifies the maximum number of columns to be displayed in a view.	300	Graph, Pivot Table

Table 19–1 (Cont.) Common Elements for Manually Changing the Display of Data in

Element	Description	Default Value	Applicable Views
MaxVisibleRows	Specifies the maximum number of rows to be displayed in a view. For tables and pivot tables, specifies the number of rows that is displayed on the tooltip for the **Display Maximum Rows per Page** paging control button.	500	Graph, Pivot Table, Table
MaxVisiblePages	Specifies the maximum number of page choices (or pages in PDF) to be displayed in a view.	1000	Graph, Pivot Table, Table
MaxVisibleSections	Specifies the maximum number of sections to be displayed in a view. This element does not apply when a slider is in place for a graph. The SectionSliderDefault and SectionSliderLimit elements apply to limit section values when a slider is in place. See Table 19–2.	25	Graph, Pivot Table, Table
JavaHostReadLimitInKB	Specifies the maximum amount of data that is sent to the browser for a single graph.	4096	Graph

19.3.2 Manually Configuring for Graphs and Gauges

You can configure various options that change the display of graphs, including funnel graphs, and gauges. These views types are also affected by the settings that are described in Section 19.3.1, "Manually Configuring for Data in Views."

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings that change the display of graphs and gauges:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"

2. Search for the Charts sections, in which you must add the elements that are described in Table 19–2.

3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <Views>
    <Charts>
     <EmbedFonts>True</EmbedFonts>
     <SectionSliderDefault>5</SectionSliderDefault>
     <SectionSliderLimit>10</SectionSliderLimit>
     <DefaultImageType>png</DefaultImageType>
     <FlashCodeBase>\\CORPORATE\Download\Flash</FlashCodeBase>
     <FlashCLSID>E38CDB6E-BA6D-21CF-96B8-432553540000</FlashCLSID>
    </Charts>
  </Views>
</ServerInstance>
```

4. Save your changes and close the file.

5. Restart Oracle Business Intelligence.

Table 19–2 Elements for Configuring Graphs and Gauges

Element	Description	Default Value
EmbedFonts	See Section 19.3.2.1, "Configuring Fonts for Graphs" for details.	False
SectionSliderDefault	Specifies the default number of values that can be displayed on a section slider bar. A section slider displays members of one or more attribute or hierarchical columns as values on a rectangular bar and provides mechanisms to select a value. For more information on defining section sliders in graphs and gauges, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.	5
SectionSliderLimit	Specifies the maximum number of values that can be displayed on a section slider bar.	10
DefaultImageType	Specifies the image type to use when displaying a live graph as opposed to a static graph that is displayed in PDF files, during downloads and previews, and so on. The default is Flash. The other choices are svg (W3C Scalable Vector Graphics) and png (W3C Portable Network Graphics). Ensure that you use lowercase letters when specifying the value. Flash and SVG images provide the greatest degree of interaction because they support mouse-over behaviors (such as pop-up data labels), navigation, and drilling.	flash
FlashCodeBase	Specifies the name of the source for downloading the Flash plug-in. The default download source for the Flash plug-in is the vendor's Web site. In some organizations, users are instructed to download the latest Flash software from a corporate location instead of the vendor's Web site. You can modify the setting to point to another location that holds the Flash code base. Then, when users view a graph and a newer version of the Flash software is available on the corporate server, they can be prompted to download the newer version.	vendor's Web site
FlashCLSID	Specifies a custom global identifier (clsid) property for downloading Flash. After modifying the Flash download directory using the FlashCodeBase element, you can enable a download prompt by creating a new classID for the Flash ActiveX control to add a custom global identifier property. You can obtain the current global identifier property from any computer where Oracle BI Presentation Services graphing is being used. (The global identifier property used by Oracle Business Intelligence is D27CDB6E-AE6D-11CF-96B8-444553540000.) The custom global identifier property must contain the same number of characters and dashes as the global identifier used in the default Flash ActiveX control. You should test flash graphs independent of Oracle Business Intelligence to ensure that they function with the custom global identifier property.	NA

19.3.2.1 Configuring Fonts for Graphs

You can do one or both of the following tasks to configuring fonts for graphs:

- Set the embed fonts element
- Deliver font files for printing

19.3.2.1.1 Setting the Embed Fonts Element By default, graphs rely on users to have the appropriate device fonts installed on their system to display multi-lingual text in the graphs. When users enable rotation on O1 axis labels, the graphs can look unattractive at certain angles. The labels appear obscured without any anti-aliasing. You can set the EmbedFonts element to True to specify the use of embedded fonts instead of device fonts, which resolves this display issue.

Be aware that the use of embedded fonts can cause a loss of fidelity. Whenever end users select fonts, they see the Oracle-licensed Albany WT plain fonts by default. Because the graphing engine does not provide embedded fonts for Chinese, Japanese, and Korean locales, users with those locales might obtain unattractive results for label rotation.

19.3.2.1.2 Delivering Font Files for Printing If you plan to print graphs in bi-directional languages to PDF or graphs in Chinese, Japanese, or Korean to PNG images, then you must deliver required font files (.TTF) as follows:

- To print graphs in bi-directional languages to PDF, you must deliver the Albany family of fonts to this Java Run-time Environment (JRE) directory:

  JAVA.HOME/lib/fonts

  where JAVA.HOME is the directory name as specified by the "java.home" system property.
- To print graphs in Chinese, Japanese, or Korean to PNG images, you must deliver the font file that contains all the needed glyphs to this JRE directory:

  lib/fonts/fallback

For more information on font configuration files, see your Java documentation.

19.3.3 Manually Changing Alternating Bar Color

This section provides information on modifying message files to perform a specific task for views. For details on working with message files, including the directories that hold them, see Section 22.5, "Customizing the User Interface Using XML Message Files."

Both tables and pivot tables can have colored bars on alternating lines. Such formatting is sometimes called "green bar styling," and the default color for these alternating bars is green. For pivot tables, content designers can control formatting features when editing tables and pivot tables, including whether alternating bar color is enabled.

As the administrator, you can change the default color for alternating bars, by editing a style configuration file. To change the color, edit the views.css file in the b_mozilla_4 folder, as shown in the following list. Change the six-digit hexadecimal color value to a new color value.

- Tables use the CSS selector:

  .ECell (for even-numbered rows)

  .OCell (for odd-numbered) rows.

- Pivot tables use the CSS selector:

  .PTE (for odd-numbered rows)

The option for enabling the alternating bars is in the Edit View dialog and is labeled **Enable alternating row "green bar" styling**. If you change the color of the bars, then you might also want to change the label to indicate the color that you have set.

To change the label in the dialog for both the table and pivot table, open the tableviewmessages.xml file and find this entry:

```
WebMessageName = "kmsgTableViewEnableGreenbarReporting"
```

Copy the entry and the text line below it to a custom messages file in the custom messages folder, and change the text line appropriately. For example:

```
WebMessageName = "kmsgTableViewEnableGreenbarReporting"
<TEXT>Enable alternating row "RED bar" styling</TEXT>"
```

19.4 Manually Changing Presentation Settings

You can configure settings that change the display of dashboards and presentation settings, as described in the following sections:

- Section 19.4.1, "Manually Changing Presentation Setting Defaults"
- Section 19.4.2, "Configuring Links to Dashboard Pages"
- Section 19.4.3, "Configuring an Alternate Toolbar for Oracle BI Publisher"
- Section 19.4.4, "Modifying the Table of Contents for PDF Versions of Briefing Books"

19.4.1 Manually Changing Presentation Setting Defaults

In addition to the presentation settings that you can change in Fusion Middleware Control, other settings can be changed manually. Use various elements in the instanceconfig.xml file to change these settings.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually change additional presentation setting defaults:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the sections in which you must add the elements that are described in Table 19–3.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example.

```
<ServerInstance>
  <Dashboard>
    <DefaultName>Templates</DefaultName>
  </Dashboard>
  <Prompts>
      <MaxDropDownValues>50</MaxDropDownValues>
  </Prompts>
  <BriefingBook>
      <MaxFollowLinks>6</MaxFollowLinks>
  </BriefingBook>
</ServerInstance>
```

Note that this example does not include parameters that might exist in the file, but that are centrally managed by Fusion Middleware Control and cannot be changed manually.

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.
6. See Section 19.4.2, "Configuring Links to Dashboard Pages" for information on the Bookmarks, MaxAgeMinutes, EnableBookmarkURL, and EnablePromptedURL elements.
7. See Section 19.4.3, "Configuring an Alternate Toolbar for Oracle BI Publisher" for information on the ReportingToolbarMode element.

Table 19–3 Elements for Manually Changing Additional Presentation Setting Defaults

Element	Description	Default Value
DefaultName	Specifies the name to be used for dashboards that contain dashboard template pages and to override the path in which Oracle BI EE searches for dashboard template pages. By default, Oracle BI EE searches for dashboard template pages in dashboards named "default" in subfolders under /Shared Folders.	default
MaxDropDownValues	Specifies the maximum number of choices to display in the following locations: ■ In choice lists in dashboard prompts. ■ In the Available list of the Select Values dialog that is displayed when you click the **Search** link in a prompt. Click the **More** link to display additional choices. ■ In the Available list of the Select Values dialog when you perform a search in that dialog. For information about prompts and searching, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.	256
MaxFollowLinks	Specifies the default value for the maximum number of navigation links to follow in a briefing book. A briefing book navigation link is a type of link that can be added to a dashboard using the Dashboard Builder. The default value for this element is 5; the minimum is 1; and the maximum is 10. If you plan to download briefing books to PDF format, then do not set the value of this element to a number greater than 9 because of the table of contents limitation of nine links. For information about the table of contents, see Section 19.4.4, "Modifying the Table of Contents for PDF Versions of Briefing Books." For information about working with briefing books, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.	5

19.4.2 Configuring Links to Dashboard Pages

Users can create links to dashboard pages. This allows them, for example, to save a link as a bookmark or to copy and send a link to other users in e-mail. A bookmark is a hidden object in the Oracle BI Presentation Catalog (under the /system/bookmarks folder) that captures the state of a dashboard page. It is created when a user creates a bookmark link to the page. You can use the following elements to configure the creation of these links:

- EnableBookmarkURL: Use this element to specify whether to show the **Create Bookmark Link** option on the Page Options menu, which allows users to create bookmark links to dashboard pages:
  - **True —** Shows the **Create Bookmark Link** option. (Default)

    If drilling in an analysis that has been set to replace a dashboard with the new results (rather than show the new results directly in the dashboard), then the **Create Bookmark Link** is displayed as a link below the new results rather than as an option on the Page Options menu.
  - **False —** Does not show the **Create Bookmark Link** option.
- EnablePromptedURL: Use this element to specify whether to show the **Create Prompted Link** option on the Page Options menu, which allows users to create prompted links to dashboard pages:
  - **True —** Shows the **Create Prompted Link** option. (Default)
  - **False —** Does not show the **Create Prompted Link** option.
- MaxAgeMinutes: Use this element within the Bookmarks element to specify that bookmarks older than the specified number of minutes are removed. The default is 43200 minutes, which corresponds to 30 days.

  Note that every time a bookmark is accessed, the expiration timer is reset. This resetting means that if a bookmark is accessed frequently, it might never be removed. Setting the value to 0 means that the bookmark is saved for 0 minutes (and does not mean that it does not expire). You cannot set bookmarks to never expire. If you want bookmarks to last for a long time, then set the value to a large number of minutes and access the bookmarks within the allotted number of minutes.

The following entry is an example of these settings:

```
<ServerInstance>
  <Dashboard>
    <EnableBookmarkURL>true</EnableBookmarkURL>
    <EnablePromptedURL>true</EnablePromptedURL>
  </Dashboard>
  <Cache>
      <Bookmarks>
          <MaxAgeMinutes>43200</MaxAgeMinutes>
      </Bookmarks>
  </Cache>
</ServerInstance>
```

19.4.3 Configuring an Alternate Toolbar for Oracle BI Publisher

When you include a BI Publisher report on a dashboard, you generally allow that report to participate as a recipient of the dashboard state by passing in dashboard context to that report using core dashboard prompts. For scenarios that do not require passing of context to or from the BI Publisher report to the larger dashboard-based analytic application, you can display a variant of the default BI Publisher toolbar, which exposes the underlying parameter prompts of that BI Publisher report. Within that frame, a user can then pass in parameters to a single BI Publisher report.

This approach can be confusing to the user as any other dashboard prompts on the page do not contribute to the BI Publisher report, which also does not participate in passing context back to the rest of the application. Changes to the BI Publisher toolbar are also applied globally for all BI Publisher reports that are embedded in dashboards across the entire Presentation Services instance.

Use the ReportingToolbarMode element to affect how BI Publisher reports are embedded in Oracle BI EE. You configure the alternate BI Publisher toolbar by setting the element's value to 6. Remove the ReportingToolbarMode element to revert to the default toolbar behavior, or set it to the default value of 1.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually configure an alternate toolbar for BI Publisher:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"

2. Locate the AdvancedReporting section in which you must add the ReportingToolbarMode element.

3. Include the element and its ancestor elements as appropriate, as shown in the following example.

   ```
   <ServerInstance>
     <AdvancedReporting>
         <ReportingToolbarMode>6</ReportingToolbarMode>
     </AdvancedReporting>
   </ServerInstance>
   ```

   The following list describes the element values:

   - 1 = Does not display the toolbar.
   - 2 = Displays the URL to the report without the logo, toolbar, tabs, or navigation path.
   - 3 = Displays the URL to the report without the header or any parameter selections. Controls such as Template Selection, View, Export, and Send are still available.
   - 4 = Displays the URL to the report only. No other page information or options are displayed.
   - 6 = Displays the BI Publisher toolbar to display the parameter prompts of the BI Publisher report

4. Save your changes and close the file.

5. Restart Oracle Business Intelligence.

19.4.4 Modifying the Table of Contents for PDF Versions of Briefing Books

The PDF version of a briefing book contains a table of contents that is automatically generated. It contains an entry for each dashboard page, analysis, and report in the briefing book. See *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for information on the table of contents.

The default template for the table of contents, toc-template.rtf, is located in the *ORACLE_INSTANCE*\config\OracleBIPresentationServicesComponent\coreapplication_obis*n* directory. You can modify the toc-template.rtf file to accommodate the needs of your organization.

19.5 Controlling Access to Saved Customization Options in Dashboards

This section provides an overview of saved customizations and information about administering saved customizations. It contains the following topics:

- Section 19.5.1, "Overview of Saved Customizations in Dashboards"
- Section 19.5.2, "Administering Saved Customizations"
- Section 19.5.3, "Permission and Privilege Settings for Creating Saved Customizations"
- Section 19.5.4, "Example Usage Scenario for Saved Customization Administration"

19.5.1 Overview of Saved Customizations in Dashboards

Saved customizations allow users to save and view later dashboard pages in their current state with their most frequently used or favorite choices for items such as filters, prompts, column sorts, drills in analyses, and section expansion and collapse. By saving customizations, users need not make these choices manually each time that they access the dashboard page.

Users and groups with the appropriate permissions and dashboard access rights can perform the following activities:

- Save various combinations of choices as saved customizations, for their personal use or use by others.
- Specify a saved customization as the default customization for a dashboard page, for their personal use or use by others.
- Switch between their saved customizations.

You can restrict this behavior in the following ways:

- Users can view only the saved customizations that are assigned to them.
- Users can save customizations for personal use only.
- Users can save customizations for personal use and for use by others.

For information about end users and saved customizations with dashboards, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

19.5.2 Administering Saved Customizations

This section describes the privileges and permissions that are required to administer saved customizations. It also describes the relevant portions of the Oracle BI Presentation Catalog that relate to storing and administering saved customizations.

19.5.2.1 Privileges for Saved Customizations

In Oracle BI Presentation Services Administration, the following privileges in the Dashboards area, along with permission settings for key dashboard elements, control whether users or groups can save or assign customizations:

- Save Customizations
- Assign Default Customizations

You can set neither privilege, one privilege, or both privileges for a user or group, depending on the level of access desired. For example, a user who has neither privilege can view only the saved customization that is assigned as his or her default customization.

19.5.2.2 Permissions for Saved Customizations

This section describes the permissions that are required for users to administer saved customizations of dashboard pages, and the relevant portions of the Oracle BI Presentation Catalog structure for setting permissions on shared and personal saved customizations.

19.5.2.2.1 Assigning Permissions to Dashboards You set permissions for dashboards and pages, such as Full Control or No Access, in the Permission dialog in Oracle BI EE. You assign these permissions in the same manner as for other objects in the catalog.

19.5.2.2.2 Assigning Permissions for Customizations on a Dashboard Page You set permissions for working with saved customizations on a particular dashboard page in the Dashboard Properties dialog, which is available in the Dashboard Builder. After selecting a page in the list in the dialog, click one of the following buttons:

- **Specify Who Can Save Shared Customizations** displays the Permission dialog in which you specify who can save shared customizations for that dashboard page.
- **Specify Who Can Assign Default Customizations** displays the Permission dialog in which you specify who can assign default customizations for that dashboard page.

Catalog objects and permissions scenarios are described in the following sections.

19.5.2.2.3 Catalog Folder Structure for Saved Customizations In addition to the privileges that you set in Oracle BI Presentation Services Administration, the level of control that users and groups have over saved customizations depends on their access rights to key elements. For example, users and groups that can create and edit underlying dashboards, save dashboard view preferences as customizations, and assign customizations to other users as default customizations require Full Control permission to the key elements in shared storage, while users and groups that can view only their assigned default saved customizations need only View access to the key elements in shared storage.

Key elements in the catalog include the following folders:

- Shared Storage Folders.

  Shared storage folders for dashboards are typically located within the Dashboards sub-folder of a parent shared folder. Dashboards are identified by their assigned names. You can save a dashboard anywhere in the Oracle BI Presentation Catalog. If you save a dashboard within a subfolder called "Dashboards", then that dashboard's name is displayed in the list of dashboards that is displayed from the Dashboards link in the global header.

  Permission settings control access to a specific dashboard for editing. Typically, if permissions are inherited down to the _selections and Dashboards sub-folders, then users who can edit dashboards can also save customizations and set defaults. Access to a specific dashboard folder controls whether a user or group can edit the dashboard.

The _selections folder contains a page identifier folder for each dashboard page. Shared saved customizations are located within this folder. Access to the page identifier folder controls whether a user or group can display, save, or edit customizations for that page.

The _defaults folder within a _selections folder contains assigned default customizations. Each group that has an assigned default is displayed here. Access to this folder controls whether a user or group can assign defaults.

- Personal Storage Folders.

  Within a user's personal folder, the _selections folder contains an individual user's saved customizations. Like the shared _selections folder, a personal _selections folder contains a page identifier folder for each dashboard page. The page identifier folder contains personal saved customizations and a _defaultlink file that specifies a user's preference for the personal defaulted customization.

  A personal saved customization default overrides an assigned shared customization default.

> **Note:** If a dashboard page with saved customizations is deleted, then the saved customizations are also deleted from the catalog. If the underlying dashboard structure changes such that a saved customization is no longer valid when a user accesses it, then the default content is displayed on the dashboard.

19.5.3 Permission and Privilege Settings for Creating Saved Customizations

Table 19–4 describes typical user roles and specific permission settings that can be granted to users for creating saved customizations. The folder names listed in the Permission and Privilege Settings column are described in the preceding section.

Table 19–4 User Roles and Permission Settings for Saved Customizations

User Role	Permission and Privilege Settings
Power users such as IT users who must perform the following tasks: ■ Create and edit underlying dashboards. ■ Save dashboard view preferences as customizations. ■ Assign customizations to other users as default customizations.	In the Shared section of the catalog, requires Full Control permission to the following folders: ■ dashboard_name ■ _selection ■ _defaults Typically, no additional privileges must be assigned.
Technical users such as managers who must perform the following tasks: ■ Save customizations as customizations for personal use. ■ Save customizations for use by others. Users cannot create or edit underlying dashboards, or assign view customizations to others as default customizations.	In the Shared section of the catalog, requires View permission to the following folders: ■ dashboard_name In the Shared section of the catalog, requires Modify permission to the following folders: ■ _selections ■ _defaults Typically, no additional privileges must be assigned.

Table 19–4 (Cont.) User Roles and Permission Settings for Saved Customizations

User Role	Permission and Privilege Settings
Everyday users who must save customizations for personal use only.	In Oracle BI Presentation Services Administration, requires the following privilege to be set: ■ Save Customizations In the dashboard page, requires that the following option is set: ■ Allow Saving Personal Customizations In the catalog, no additional permission settings are typically required.
Casual users who must view only their assigned default customization.	In the Shared section of the catalog, the user needs View permission to the following folders: ■ dashboard_name ■ _selections ■ _defaults In the catalog, no additional permission settings are typically required.

19.5.4 Example Usage Scenario for Saved Customization Administration

Depending on the privileges set and the permissions granted, you can achieve various combinations of user and group rights for creating, assigning, and using saved customizations.

For example, suppose a group of power users cannot change dashboards in a production environment, but they are allowed to create saved customizations and assign them to other users as default customizations. The following permission settings for the group are required:

- Open access to the dashboard, using the Catalog page.
- Modify access to the _selections and _defaults subfolders within the dashboard folder in the Oracle BI Presentation Catalog, which you assign using the Dashboard Properties dialog in the Dashboard Builder. After selecting a page in the list in the dialog, click **Specify Who Can Save Shared Customizations** and **Specify Who Can Assign Default Customizations**.

19.6 Blocking Analyses in Answers

You might want to block specific analyses, such as requiring content designers to include certain columns with others, or requiring filters when certain columns are requested. Answers includes an API that you can use to block queries based on the criteria specified in the analysis or based on formulas in the analysis. You can access the API using JavaScript to check conditions and validate analyses.

This section contains the following topics:

- Section 19.6.1, "Storing JavaScript Files"
- Section 19.6.2, "Blocking Analyses Based on Criteria"
- Section 19.6.3, "Blocking Analyses Based on Formula"
- Section 19.6.4, "Validation Helper Functions"

19.6.1 Storing JavaScript Files

This section explains how to use JavaScript to check conditions and validate analyses. You write your own JavaScript programs for performing these tasks and other similar ones. Oracle BI EE does not install any JavaScript programs. As you write JavaScript programs, you can store them in the following directory:

ORACLE_INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\analyticsRes

To place JavaScript programs in a directory other than this one, then you can do so, if you specify the full path name in the code that calls the program. For example, you can use code such as the following:

```
<script type="text/javascript" src="http://example/mydir/myblocking.js" />
```

19.6.2 Blocking Analyses Based on Criteria

When a user attempts to execute an analysis that your code blocks, you can display an error message, and the analysis is not executed. The answerstemplates.xml file includes a message named kuiCriteriaBlockingScript that can be overridden to either define or include JavaScript that defines a validateAnalysisCriteria function. By default, this message contains a function that always returns True. It should be overridden using the procedures that are described in Section 22.4, "Customizing Language Selections and Other Components."

Answers calls your validateAnalysisCriteria function when the user tries to execute the analysis. The function can return True if the analysis is not blocked, or False, or a message if the analysis is blocked. If a message or a value other than False is returned, then the message is displayed in a popup window. In either case, the query is blocked.

The following code example shows the blocking of a query. First, place the following XML code in the answerstemplates.xml file.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="QueryBlocking" table="Messages">
   <WebMessage name="kuiCriteriaBlockingScript" translate="no">
      <HTML>
         <script type="text/javascript" src="fmap:myblocking.js" />
      </HTML>
   </WebMessage>
   </WebMessageTable>
</WebMessageTables>
```

This XML code calls a JavaScript program called myblocking.js. Ensure that you place this file in the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\analyticsRes directory. The following is sample code for the myblocking.js program.

```
// This is a blocking function. It makes sure users pick what I want them to.
function validateAnalysisCriteria(analysisXml)
{
   // Create the helper object
   var tValidator = new CriteriaValidator(analysisXml);
   // Validation Logic
   if (tValidator.getSubjectArea() != "Paint")
      return "Why don't you try Paint?";
   if (!tValidator.dependentColumnExists("Markets","Region","Markets","District"))
   {
      // If validation script notifies user, then return false
      alert("Region and District go so well together, do you think?");
      return false;
   }
   if (!tValidator.dependentColumnExists("Sales Measures","","Periods","Year"))
   return "You picked a measure so pick Year!";
   if (!tValidator.filterExists("Sales Measures","Dollars"))
   return "Maybe filter on Dollars?";
   if (!tValidator.dependentFilterExists("Markets","Market","Markets"))
   return "Since you're showing specific Markets, filter the markets.";
   var n = tValidator.filterCount("Markets","Region");
   if ((n <= 0) || (n > 3))
      return "Select 3 or fewer specific Regions";
   return true;
}
```

If you do not override the function using the template as described previously, or if the function returns anything other than False, then the criteria is considered to be valid and the analysis is issued. The criteria is validated using this same mechanism for preview and save operations as well.

After making this change, either stop and restart the server for Oracle BI Presentation Services, or click the **Reload Files and Metadata** link on the Administration page.

19.6.3 Blocking Analyses Based on Formula

Answers provides a hook that lets you incorporate a JavaScript validation function that is called from Answers when a content designer enters or modifies a column formula. If the call fails and returns a message, then Answers displays the message and cancels the operation. Additionally, helper functions are available so the query blocking function can check for filters, columns, and so on, rather than traversing the Document Object Model (DOM) manually. (The DOM is a way of describing the internal browser representation of the HTML UI page that is currently being displayed in Answers.) For more information on the helper functions, see Section 19.6.4, "Validation Helper Functions."

The criteriatemplates.xml file includes a message named kuiFormulaBlockingScript that can be overridden to include JavaScript that defines a validateAnalysisFormula function. By default, this message contains a function that always returns True.

Answers calls validateAnalysisFormula before applying changes made by the content designer. If the function returns True, then the formula is accepted. If the function returns False, then the formula is rejected. Otherwise, the return value from the function is displayed in the message area beneath the formula, as it does currently when an invalid formula is entered.

The content designer has the option to click **OK** to ignore the error. To display your own alert and allow the content designer to continue, your function should return True. To block the query, return False or a message. Your function should investigate the formula passed to it using a JavaScript string and regular expression techniques for validation.

The following code example shows a sample custom message.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="QueryBlocking" table="Messages">
      <WebMessage name="kuiFormulaBlockingScript" translate="no">
         <HTML>
```

```
            <script type="text/javascript" src="fmap:myblocking.js" />
         </HTML>
      </WebMessage>
   </WebMessageTable>
</WebMessageTables>
```

The following code example shows blocking based on the formula entered.

```
// This is a formula blocking function. It makes sure the user does not enter an
unacceptable formula.
function validateAnalysisFormula(sFormula, sAggRule)
{
   // we do not allow the use of concat || in our formulas
   var concatRe = /\|\|/gi;
   var nConcat = sFormula.search(concatRe);
   if (nConcat >= 0)
      return "You used concatenation (character position " + nConcat + "). That is
not allowed.";
   // no case statements
   var caseRe = /CASE.+END/gi;
   if (sFormula.search(caseRe) >= 0)
      return "Do not use a case statement.";
   // Check for a function syntax: aggrule(formula) aggrule should not contain a
'.'
   var castRe = /^\s*\w+\s*\(.+\)\s*$/gi;
   if (sFormula.search(castRe) >= 0)
      return "Do not use a function syntax such as RANK() or SUM().";
    return true;
}
```

After making this change, either stop and restart the server for Oracle BI Presentation Services, or click the **Reload Files and Metadata** link on the Administration page.

19.6.4 Validation Helper Functions

These functions are defined within a JavaScript file named answers/queryblocking.js. Table 19–5 contains the list of helper functions and their descriptions.

Table 19–5 Validation Helper Functions

Validation Helper Function	Description
CriteriaValidator.getSubjectArea()	Returns the name of the subject area referenced by the analysis. It generally is used in a switch statement within the function before doing other validation. If the analysis is a set-based criteria, then it returns null.
CriteriaValidator.tableExists(sTable)	Returns True if the specified folder (table) has been added to the analysis by the content designer, and False if the folder was not added.
CriteriaValidator.columnExists(sTable, sColumn)	Returns True if the specified column has been added to the analysis by the content designer, and False if the column was not added.

Table 19–5 (Cont.) Validation Helper Functions

Validation Helper Function	Description
CriteriaValidator.dependentColumnExists(sCheckTable, sCheckColumn, sDependentTable, sDependentColumn)	Checks to ensure that the dependentColumn exists if the checkColumn is present. It returns True if either the checkColumn is not present, or the checkColumn and the dependent column are present. If checkColumn and dependentColumn are null, then the folders are validated. If any column from checkTable is present, then a column from dependentTable must be present.
CriteriaValidator.filterExists(sFilterTable, sFilterColumn)	Returns True if a filter exists on the specified column, and False if no filter is present.
CriteriaValidator.dependentFilterExists(sCheckTable, sCheckColumn, sFilterTable, sFilterColumn)	Checks to ensure that the dependentFilter exists if the checkColumn is present in the projection list. It returns True if either the checkColumn is not present, or the checkColumn and the dependent filter are present.
CriteriaValidator.filterCount(sFilterTable, sFilterColumn)	Returns the number of filter values that are specified for the given logical column. If the filter value is "equals," "null," "notNull," or "in," then it returns the number of values chosen. If the column is not used in a filter, then it returns zero. If the column is prompted with no default, then it returns -1. For all other filter operators (such as "greater than," "begins with," and so on) it returns 999, because the number of values cannot be determined.

19.7 Specifying View Defaults for Analyses and Dashboards

You can control certain aspects of the initial state of new views that are added to an analysis and of new objects that are added to a dashboard page. For example, you can add a default footer to new analyses and set defaults for dashboard sections. You control these aspects by customizing the appropriate XML message files to override the default values that are specified during installation.

19.7.1 XML Message Files for View Defaults

This section describes the XML message files to customize to override the view defaults distributed with Oracle BI Presentation Services.

For analyses, the file answerstemplates.xml includes a message named kuiCriteriaDefaultViewElementsWrapper from within kuiAnswersReportPageEditorHead. This message includes two additional messages, kuiCriteriaDefaultViewElements, in which you can define default values, and kuiCriteriaDefaultViewElementsMask, in which masks are defined. The mask XML message is protected and you cannot modify its contents.

The wrapper message adds the combined XML into a JavaScript variable, kuiDefaultViewElementsXML, that is used to apply the new default values.

For dashboards, the file dashboardtemplates.xml includes a message named kuiDashboardDefaultElementsWrapper that adds XML into a JavaScript variable named kuiDefaultDashboardElementsXML for use within the dashboard editor.

Note: For information about the core tasks that are required to customize XML message files, see Section 22.5, "Customizing the User Interface Using XML Message Files." The examples in this section assume that you are familiar with this information.

19.7.2 Examples of Customizing Default Values for Analyses and Dashboards

The following sections provide examples of customizing default values:

- Section 19.7.2.1, "Adding a Default Header or Footer to New Analyses"
- Section 19.7.2.2, "Preventing Auto-Previewing of Results"
- Section 19.7.2.3, "Setting Defaults for Analyses in the Compound Layout"
- Section 19.7.2.4, "Changing Dashboards Section Defaults"
- Section 19.7.2.5, "Specifying Dashboard Page Defaults Including Headers and Footers"
- Section 19.7.2.6, "Including Links with Analyses on Dashboards"

To cause these customizations to take effect, either stop and restart the server for Oracle BI Presentation Services, or click the **Reload Files and Metadata** link on the Administration page.

19.7.2.1 Adding a Default Header or Footer to New Analyses

You can specify that default headers and footers are displayed on all new analyses. For example, footers can contain messages such as a confidentiality notice, the company's name, and so on. You can specify a default header or footer by creating an XML message that specifies the text and formatting to apply.

The following XML code example creates a footer that contains the text "Acme Confidential" in bold, red letters.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">
<WebMessage name="kuiCriteriaDefaultViewElements" translate="no"><HTML>
   <view signature="compoundView" >
      <pageProps pageSize="a4">
         <pageFooter showOnDashboard="true" show="true">
            <zone type="top"><caption>[b]Acme Confidential[/b]</caption>
            <displayFormat fontColor="#FF0000"/></zone>
         </pageFooter>
      </pageProps>
   </view>
</HTML>
</WebMessage>
   </WebMessageTable>
</WebMessageTables>
```

19.7.2.2 Preventing Auto-Previewing of Results

The results of an analysis are displayed when editing views of data. If you prefer that the content designer explicitly asks to view the results, then you can create an XML message that specifies that auto-preview should be disabled when new views are created. The content designer can still click the **Display Results** link to view the results when editing a view.

The following XML code example disallows the auto-previewing of results when working with a view in Answers.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">
<WebMessage name="kuiCriteriaDefaultViewElements" translate="no"><HTML>
   <view signature="tableView" showToolBar="true" showHeading="true />
   <view signature="pivotTableView" autoPreview="false" />
   <view signature="titleView" autoPreview="false" />
   <view signature="viewSelector" autoPreview="false" />
   <view signature="htmlviewnarrativeView" autoPreview="false" />
   <view signature="tickerview" autoPreview="false" />
   <view signature="htmlview" autoPreview="false" />
</HTML>
</WebMessage>
   </WebMessageTable>
</WebMessageTables>
```

19.7.2.3 Setting Defaults for Analyses in the Compound Layout

The results of a newly formed analysis are displayed as a title view followed by either a table or pivot table in a compound layout. A table is created if the analysis contains only attribute columns, and a pivot table is created if the analysis contains at least one hierarchical column.

You can create an XML message that specifies that the compound view should default to a different assemblage of views, such as a narrative followed by a graph. The content designer can still add and rearrange views within the compound layout.

The following XML code example sets the default compound layout to a narrative followed by a graph.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">
<WebMessage name="kuiCriteriaDefaultViewElements" translate="no"><HTML>
   <view signature="compoundView" >
      <cv signature="narrativeView" />
      <cv signature="dvtchart" />
   </view>
</HTML>
</WebMessage>
   </WebMessageTable>
</WebMessageTables>
```

19.7.2.4 Changing Dashboards Section Defaults

By default, the results of drilling in a dashboard are displayed on a new page, section names are not displayed in the dashboard, and sections can be expanded and collapsed. You can change these default values by creating an XML message that specifies that new default values for the dashboard section. A content designer who edits the dashboard can still modify this behavior using the options within the dashboard editor.

The following XML code example makes section heads visible, enables drilling, and does not allow sections to collapse.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">
<WebMessage name="kuiDashboardDefaultElements" translate="no"><HTML>
```

```
   <element signature="dashboardSection" drillInline="true" showHeading="true"
collapsible="false" />
</HTML>
</WebMessage>
   </WebMessageTable>
</WebMessageTables>
```

19.7.2.5 Specifying Dashboard Page Defaults Including Headers and Footers

By default, dashboards are printed without headers or footers and in a portrait orientation. If you prefer that newly added dashboard pages default to having a custom header and footer and print in landscape orientation, then you can create an XML message that specifies these characteristics. A content designer who edits the dashboard can still modify this behavior using the options within the dashboard editor.

The following XML code example adds a custom header and footer to a dashboard page and specifies landscape orientation.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">
<WebMessage name="kuiDashboardDefaultElements" translate="no"><HTML>
<element signature="dashboardPage" personalSelections="false">
     <pageProps orientation="portrait" printRows="all" pageSize="a4">
         <pageHeader showOnDashboard="true" show="true">
            <zone type="top"><caption>[b]Acme is Cool[/b]</caption>
            <displayFormat fontSize="9pt" hAlign="center"
fontColor="#FFFFFF" backgroundColor="#000000"/></zone>
         </pageHeader>
         <pageFooter showOnDashboard="true" show="true">
            <zone type="top"><caption>[b]CONFIDENTIAL
@{timeCreated[mm/dd/yy]}[/b]</caption>
            <displayFormat fontSize="7.5pt" hAlign="center"
fontColor="#999999" borderColor="#CC99CC" fontStyle="italic"
borderPosition="all" borderStyle="single"/></zone>
         </pageFooter>
       </pageProps>
   </element>
</HTML>
</WebMessage>
   </WebMessageTable>
  </WebMessageTables>
```

19.7.2.6 Including Links with Analyses on Dashboards

By default, the results of embedded analyses are displayed within the dashboard without including any links. If you prefer that newly added analyses default to having **Edit** and **Refresh** links, for example, then you can create an XML message that specifies that the analysis should behave this way. A content designer who edits the dashboard can still modify this behavior using the options within the Dashboard Builder.

In the XML message file, the links attribute can contain any combination of the letters c, d, e, f, g, m, and r to add the indicated link, as shown in the following list that provides the letter and corresponding link name:

- c = Copy
- d = Export
- e = Edit

- f = Print
- g = Add to Briefing Book
- m = Analyze
- r = Refresh

The following XML code example adds **Edit** and **Refresh** links to new analyses embedded in dashboards.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">
<WebMessage name="kuiDashboardDefaultElements" translate="no"><HTML>
   <element signature="reportView" display="embed" links="er" />
</HTML>
</WebMessage>
   </WebMessageTable>
</WebMessageTables>
```

19.8 Configuring for Write Back in Analyses and Dashboards

Users of a dashboard page or an analysis might have the ability to modify the data that they see in a table view. This ability is often referred to as "write back." As the administrator, you assist the content designer in configuring write back for users.

Detailed information about write back in views is provided in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*. The following sections provide information on how you as the administrator can configure for write back:

- Section 19.8.1, "Write-Back Limitations"
- Section 19.8.2, "Configuring for Write Back"
- Section 19.8.3, "About the Write-Back Template"

19.8.1 Write-Back Limitations

Users can write back to any data source (except for an ADF data source) that allows the execution of SQL queries from the Oracle BI Server. As you configure for write back, keep the following limitations in mind:

- Numeric columns must contain numbers only. They should not contain any data formatting characters such as dollar signs ($), pound signs or hash signs (#), percent signs (%), and so on.
- Text columns should contain string data only.
- You can use the template mechanism only with table views and only for single-value data. The template mechanism is not supported for pivot table views or any type of view or for multiple-value data.
- All values in write-back columns are editable. When displayed in non printer friendly context, editable fields are displayed as if the user has the Write Back to Database privilege. However, when a logical column is mapped to a physical column that can change, the logical column returns values for multiple level intersections. This scenario can cause problems.
- Any field in an analysis can be flagged as a write-back field, even if it is not derived from the write-back table that you created. However you cannot successfully execute the write-back operation if the table is not write-back enabled. The responsibility for correctly tagging fields lies with the content designer.

- A template can contain SQL statements other than insert and update. The write-back function passes these statements to the database. However, Oracle does not support or recommend the use of any statements other than insert or update.
- Presentation Services performs only minimal validation of data input. If the field is numeric and the user enters text data, then Presentation Services detects that and prevents the invalid data from going to the database. However, it does not detect other forms of invalid data input (values out of range, mixed text and numeric, and so on). When the user clicks the write-back button and an insert or update is executed, invalid data results in an error message from the database. The user can then correct the faulty input. Content designers can include text in the write-back analysis to aid the user, for example, "Entering mixed alphanumeric values into a numeric data field is not allowed."
- The template mechanism is not suitable for entering arbitrary new records. In other words, do not use it as a data input tool.
- Write-back analyses do not support drill-down. Because drilling down modifies the table structure, the write-back template does not work.

> **Caution:** The template mechanism takes user input and writes it directly to the database. The security of the physical database is your own responsibility. For optimum security, store write-back database tables in a unique database instance.

19.8.2 Configuring for Write Back

Complete the following steps to configure for users to write back values to the data source.

To configure for write back:

1. Create a physical table in the database that has a column for each write-back field needed. In the table create statement, make the write-back fields non-null-able.

   > **Note:** For optimum security, store write-back database tables in a unique database instance.

2. Use the Oracle BI Administration Tool to configure the new table, as described in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.
3. Create a write-back template that specifies the SQL statements that are necessary to both insert and update values into the table that you created. For more information, see Section 19.8.3, "About the Write-Back Template."
4. Add the LightWriteback element in the instanceconfig.xml file, as described in Section 19.8.2.1, "Setting the LightWriteback Element."
5. In Oracle BI Presentation Services, grant the following write-back privileges to the appropriate users: Manage Write Back and Write Back to Database.

   For information, see Section C.2.3.2, "Setting Privileges in Oracle BI Presentation Services Administration."
6. Configure a write-back analysis, as described in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

19.8.2.1 Setting the LightWriteback Element

In order for users to write back values, you must manually add the LightWriteback element in the instanceconfig.xml file. Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually set the element for write back:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the ServerInstance section in which you must add the LightWriteback element.
3. Include the element and its ancestor elements as appropriate, as shown in the following example.

   ```
   <WebConfig>
       <ServerInstance>
           <LightWriteback>true</LightWriteback>
       </ServerInstance>
   <WebConfig>
   ```

   Note that this example does not include parameters that might exist in the file, but that are centrally managed by Fusion Middleware Control and cannot be changed manually.
4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

19.8.3 About the Write-Back Template

The write-back template is an XML-formatted file that contains SQL statements that are needed to insert and update records in the write-back table and columns that you have created. You can create multiple write-back templates, customizing each one for the fields that are used in each specific analysis. In the table view properties, you specify the name of the write-back template to use.

19.8.3.1 How Write Back Works

If a user has the Write Back to Database privilege, then the write-back fields in their analyses can display as editable fields if properly configured. If the user does not have this privilege, then the write-back fields display as normal fields. If the user types a value in an editable field and clicks the appropriate write-back button, then the application reads the write-back template to get the appropriate insert or update SQL command. It then issues the insert or update command. If the command succeeds, then it reads the record and updates the analysis. If there is an error in either reading the template or in executing the SQL command, then an error message is displayed.

The insert command runs when a record does not yet exist and the user enters new data into the table. In this case, a user has typed in a table record whose value was originally null.

The update command runs when a user modifies existing data. To display a record that does not yet exist in the physical table to which a user is writing back, you can create another similar table. Use this similar table to display placeholder records that a user can modify in dashboards.

19.8.3.2 Requirements for the Write-Back Template

The write-back template must meet the following requirements:

- To meet security requirements, you must specify the connection pool along with the SQL commands to insert and update records. These SQL commands reference the values that are passed in the write-back schema to generate the SQL statements to modify the database table. Values can be referenced either by column position (such as @1, @3) or by column ID (such as @{c1234abc}, @{c687dfg}). Column positions start numbering with 1. The use of column ID is preferred. Each column ID is alphanumeric, randomly generated, and found in the XML definition of the analysis in the Advanced tab of the Analysis editor.
- You must include both an <insert> and an <update> element in the template. If you do not want to include SQL commands within the elements, then you must insert a blank space between the opening and closing tags. For example, you must enter the element as

  ```
  <insert> </insert>
  ```

  rather than

  ```
  <insert></insert>
  ```

  If you omit the blank space, then you see a write-back error message such as "The system cannot read the Write Back Template 'my_template'".
- If a parameter's data type is not an integer or real number, then add single quotes around it. If the database does not do Commits automatically, then add the optional postUpdate node after the insert and update nodes to force the commit. The postUpdate node typically follows this example:

  ```
  <postUpdate>COMMIT</postUpdate>
  ```

- Store the write-back template files in the analyticsRes directory that the administrator has configured for static files and customer messages:

  ORACLE_INSTANCE/bifoundation/OracleBIPresentationServicesComponent/coreapplication_obips*n*/analyticsRes/customMessages

  While XML message files that affect a language-specific user interface must be localized, the XML file that is used for configuring a write-back template is usually not translated, because it is language-independent.

  In the rare cases where write-back template files must be language-dependent (for example, if a user logging in using the l_es (Spanish) locale would use a different SQL command then a user logging in using l_fr (French) locale), then the write-back template messages should exist in appropriate language directories. For information, see Section 22.5, "Customizing the User Interface Using XML Message Files."
- The write-back template files can have any name of your choosing, because the system reads all XML files in the CustomMessages folder. To ensure that write back works correctly, include in the WebMessage element of the file the name of the SQL template that you specified when you created the write-back table. You can have multiple WebMessage elements in one file, with each element specifying one SQL template.

  The following example shows the specification of the SQL template that is called "SetQuotaUseID."

```
    <WebMessage name="SetQuotaUseID">
```

19.8.3.3 Example: Write Back Template

A write-back template might resemble this example:

```
<?xml version="1.0" encoding="utf-8" ?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web/message/v1">
<WebMessageTable lang="en-us" system="WriteBack" table="Messages">
   <WebMessage name="SetQuotaUseID">
      <XML>
         <writeBack connectionPool="Supplier">
            <insert>INSERT INTO regiontypequota
VALUES(@{c0},@{c1},'@{c2}','@{c3}',@{c4})</insert>
            <update>UPDATE regiontypequota SET Dollars=@{c4} WHERE YR=@{c0} AND
Quarter=@{c1} AND Region='@{c2}' AND ItemType='@{c3}'</update>
         </writeBack>
      </XML>
   </WebMessage>
   <WebMessage name="SetQuota">
      <XML>
         <writeBack connectionPool="Supplier">
            <insert>INSERT INTO regiontypequota
VALUES(@1,@2,'@3','@4',@5)</insert>
            <update>UPDATE regiontypequota SET Dollars=@5 WHERE YR=@1 AND
Quarter=@2 AND Region='@3' AND ItemType='@4'</update>
         </writeBack>
      </XML>
   </WebMessage>
</WebMessageTable>
</WebMessageTables>
```

20 Configuring and Managing Agents

If your organization licensed Oracle BI Delivers and if you have the appropriate privileges, then you can use the agents functionality as part of a default installation with no additional configuration. This chapter describes how agents are used, what affects agents, about advanced configuration settings that affect agents, managing device types for agents, and monitoring agent sessions. For information about using agents, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

This chapter contains the following sections:

- Section 20.1, "How are Agents Used?"
- Section 20.2, "How Do Antivirus Software and Privileges Affect Agents?"
- Section 20.3, "Configuring Settings That Affect Agents"
- Section 20.4, "Managing Device Types for Agents"
- Section 20.5, "Monitoring Active Agent Sessions"

Note: If you are migrating an Oracle Business Intelligence environment to a new system, then ensure that you also migrate the Oracle Business Intelligence repository file, the Oracle BI Presentation Catalog, and the Oracle BI Scheduler tables. The Oracle BI Scheduler tables are required for agents.

See Section 8.4.5, "Diagnosing Issues with the Agents" for information on diagnostics and log files for agents.

20.1 How are Agents Used?

Agents deliver targeted analytics to users based on a combination of schedule and trigger event. Delivery can be by a variety of routes, for example to Dashboard Alerts or to e-mail.

To create an agent, Oracle Business Intelligence users (with the Create Agent privilege) define the operations that the agent is to perform. Oracle BI Presentation Services packages information such as priority, delivery devices, and user, into a job, and tells Oracle BI Scheduler when to execute the job. For information about the Oracle BI Scheduler, see Section 24.1, "What is Oracle BI Scheduler?"

20.2 How Do Antivirus Software and Privileges Affect Agents?

This section provides the following information about agents:

- Section 20.2.1, "How Does Antivirus Software Affect Agents?"
- Section 20.2.2, "What Privileges Affect Agents?"

20.2.1 How Does Antivirus Software Affect Agents?

Some antivirus software programs, such as Norton AntiVirus, enable a script-blocking feature, which tries to block all calls made by scripts to system objects (such as the Windows file system object) that the antivirus software deems unsafe.

If you launch a script as part of post-agent processing, then this antivirus feature might cause unexpected results. If you run antivirus software with a script-blocking feature on the computer where Oracle BI Scheduler is installed, then you should disable the script-blocking feature to prevent the software from unexpectedly blocking agent script calls.

20.2.2 What Privileges Affect Agents?

You access the privilege settings for agents in the Delivers section of the Manage Privileges page in Oracle BI Presentation Services Administration.

To create an agent, users must be granted the Create Agent privilege. To enable users with the Publish Agents for Subscription privilege, which provides the ability to change or to delete an agent, you must grant them the Modify permission to the shared agent objects and child objects in the Oracle BI Presentation Catalog. For information, see Section C.1.3, "How Are Permissions and Privileges Assigned to Users?"

> **Note:** If the Oracle BI Server is configured to authenticate users through database logons, then impersonation is permitted until the number of associated variables exceeds one (for example, when session variables other than USER are associated with the initialization block). If the number of associated variables exceeds one, then the impersonated user does not have the password to log in to the database and to fill the other session variables. Agents work with database authentication, if only the initialization block that is set up for authentication in the Oracle BI Administration Tool uses a connection pool with pass-through login. That connection pool cannot be used for any other initialization block or request.
>
> For information about user authentication options, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*. For information about pass-through login, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

20.3 Configuring Settings That Affect Agents

You configure settings for agents by changing values for Oracle BI Presentation Services or Oracle BI Scheduler. You configure delivery options for agents using the SA System subject area. This section contains the following topics:

- Section 20.3.1, "Manually Configuring Presentation Services Settings That Affect Agents"
- Section 20.3.2, "Manually Changing Additional Scheduler Settings that Affect Agents"

- Section 20.3.3, "What Additional Scheduler Configuration Settings Affect Agents?"
- Section 20.3.4, "Controlling Delivery Options for Agents"

20.3.1 Manually Configuring Presentation Services Settings That Affect Agents

Use various elements in the instanceconfig.xml file for Presentation Services to change these settings. You must apply changes to both the primary and secondary scheduler's instanceconfig.xml in a cluster.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit Presentation Services settings that affect agents:

1. Open the Presentation Services instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the elements that are described in Table 20–1.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <ps:Alerts xmlns:ps="oracle.bi.presentation.services/config/v1.1">
     <ps:Enabled>false<ps:/Enabled>
     <ps:DefaultDeliveryFormat>pdf</ps:DefaultDeliveryFormat>
  </ps:Alerts>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Presentation Services.

Table 20–1 Presentation Services Settings That Affect Agents

Element	Description	Default Value
Enabled	Specifies whether Oracle BI Delivers is enabled. Allowed values are true or false. Delivers is an optional component of Presentation Services that is enabled by default for organizations that have purchased the appropriate license. You use the Delivers component to create agents.	true
DefaultDeliveryFormat	Specifies the default format for sending e-mailed reports through an agent. For example, a content designer can create an agent to send a report every day to a development team to share how many bugs have been fixed in the past day. When the content designer creates the agent, he can specify the format of the e-mail. As the administrator, you can specify the default format that is used for such e-mails, using one of the following values: html pdf excel text	html

20.3.2 Manually Changing Additional Scheduler Settings that Affect Agents

In addition to the scheduler settings that you can change in Fusion Middleware Control, you can change other settings manually. Use various elements in the instanceconfig.xml file to change these settings. You must apply changes to both the primary and secondary scheduler's instanceconfig.xml in a cluster.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually change additional Oracle BI Scheduler settings that affect agents:

1. Open the Oracle BI Scheduler version of the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the sections in which you must add or update the elements that are described in Section 20.3.3, "What Additional Scheduler Configuration Settings Affect Agents?"
3. Include the elements and their ancestor elements as appropriate. For example the entry for Log_Dir is shown below:

```
<xs:element name="Log_Dir" type="xs:string" default="ibots" minOccurs="0">
   <xs:annotation>
    <xs:documentation xml:lang="en">
     The directory where Agent logs are stored.
    </xs:documentation>
   </xs:annotation>
</xs:element>
```

> **Note:** You cannot specify values for user name and password in the instanceconfig.xml file. Instead you specify values in Fusion Middleware Control that are stored securely within the central credentials wallet, along with all other user names and passwords.

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

20.3.3 What Additional Scheduler Configuration Settings Affect Agents?

You can change the following additional Oracle BI Scheduler configuration settings that affect agents:

- Section 20.3.3.1, "General Scheduler Configuration Settings That Affect Agents"
- Section 20.3.3.2, "Email Scheduler Configuration Settings That Affect Agents"
- Section 20.3.3.3, "Agent Scheduler Configuration Settings"

20.3.3.1 General Scheduler Configuration Settings That Affect Agents

General configuration settings include access to, and configuration of, the Scheduler back-end database, some behavior settings, and settings for secure sockets and clustering configuration. Table 20–2 describes the settings.

Table 20–2 General Scheduler Configuration Settings That Affect Agents

Element	Description	Default Value
PoolTimeout	Specifies the amount of time in minutes that a connection to the data source remains open after an operation completes. During this time, new operations use this connection rather than open a new one, up to the number specified for Maximum Connections. The time is reset after each completed connection request. Specify a value of 1 or greater.	60
NumDbConnections	Specifies the maximum number of database connections that Oracle BI Scheduler can open concurrently. Specify a value of 1 or greater. When this limit is reached, the connection request waits until a connection becomes available.	5
TABLE_JOBS	Specifies the name of a database table used to store information about scheduled jobs. **Note**: For information about modifying the database table names, see Section 25.4, "Changing Oracle BI Scheduler Table Names" and *Oracle Business Intelligence Applications Installation and Configuration Guide*.	S_NQ_JOB
TABLE_INSTANCES	Specifies the name of a database table used to store information about job instances.	S_NQ_INSTANCE
TABLE_PARAMS	Specifies the name of a database table used to store information about job parameters.	S_NQ_JOB_PARAM
TABLE_ERRMSGS	Specifies the name of a database table used to store information about job instances that do not complete successfully.	S_NQ_ERR_MSG
SchedulerScriptPath	Refers to the path where Oracle BI Scheduler-created job scripts are stored. In general, you should not add or remove scripts from this directory. By default, this field is set to *ORACLE_INSTANCE*\bifoundation\OracleBISchedulerComponent\coreapplication_obisch*n*\scripts\scheduler, for example D:\OBI11g\instances\instance1\bifoundation\OracleBISchedulerComponent\coreapplication_obisch1\scripts\scheduler.	scripts\scheduler
DefaultScriptPath	Specifies the path where user-created job scripts (not agents) are stored. If a file name is entered in the Script field when adding or modifying a job, then Oracle BI Scheduler examines the contents of this directory for the specified file. However, if a full path is given in the Script field, then this directory is not examined. By default, this field is set to *ORACLE_INSTANCE*\bifoundation\OracleBISchedulerComponent\coreapplication_obisch*n*\scripts\scheduler\common, for example D:\OBI11g\instances\instance1\bifoundation\OracleBISchedulerComponent\coreapplication_obisch1\scripts\scheduler\common.	scripts\scheduler\common
TempPath	Specifies the path where temporary files are stored during Oracle BI Scheduler execution.	NA
BulkFetchBufferSize	Used in the database gateways. Specifies the maximum size in bytes of a bulk fetch page for retrieving data from a data source.	33,792 bytes.

Table 20–2 (Cont.) General Scheduler Configuration Settings That Affect Agents

Element	Description	Default Value
LogAllSqlStmts	Occasionally you might set up the Scheduler to point to a database using a generic protocol like ODBC. This is usually done when the Target Type is not specified. When this happens, and a SQL statement fails, you must be able to determine which statement failed. Turning this setting on places the SQL statements in the Scheduler log file. Do not set this to TRUE in production mode as the overhead for this is quite high.	false
ServerPortNumber	Specifies the port number for the server. Defaults to the Oracle BI Scheduler port number.	The port number set for the Scheduler.
PurgeInstDays	Specifies the number of days after which old job instances are deleted from the back-end database automatically. To prevent old job instances from being deleted automatically, set the value to 0 (zero).	7 days
PurgeIntervalMinutes	Specifies the number of minutes in which Oracle BI Scheduler updates the tables and flags the affected rows as deleted. **Note**: Oracle BI Scheduler does not actually issue SQL DELETE statements when jobs or instances are removed, instead rows are flagged for deletion. After every *X* minutes (where *X* is defined as the value of this field), the actual SQL DELETE statements are issued.	60 minutes
MinExecThreads	Specifies the minimum number of multiple threads in the Oracle BI Scheduler thread pool that executes jobs at run time.	1
MaxExecThreads	Specifies the maximum number of multiple threads in the Oracle BI Scheduler thread pool that executes jobs at run time.	100
PauseOnStartup	Specifies that no jobs should execute when Oracle BI Scheduler starts. While Oracle BI Scheduler pauses, users can add, modify, and remove jobs. However, no jobs execute. From the **Service Management** menu, select **Continue Scheduling** to continue with regular execution.	false
CertificateFileName	Specifies the SSL Certificate File Path. This setting supports SSL.	NA
CertPrivateKeyFileName	Specifies the SSL Certificate Private Key File. This setting supports SSL.	NA
PassphraseFileName	Specifies the SSL File Containing Passphrase. This setting supports SSL.	NA
PassphraseProgramName	Specifies the SSL Program Producing Passphrase. This setting supports SSL.	NA
CertificateVerifyDepth	Specifies the SSL Certificate Verification Depth.	NA
CACertificateDir	Specifies the CA Certificate Directory. This setting supports SSL.	NA

Table 20–2 (Cont.) General Scheduler Configuration Settings That Affect Agents

Element	Description	Default Value
CACertificateFile	Specifies the CA Certificate File. This setting supports SSL.	NA
TrustedPeerDNs	Specifies the SSL Trusted Peer DNs.	NA
VerifyPeer	Specifies whether to verify the peer. This setting supports SSL.	false
CipherList	Specifies the Cipher List. This setting supports SSL.	NA
ScriptRPCPort	Specifies the port used for interprocess communication between the script processes and the Oracle BI Scheduler process. This port accepts connections only from the same computer on which Oracle BI Scheduler is running.	9707

20.3.3.2 Email Scheduler Configuration Settings That Affect Agents

Table 20–3 Email Scheduler Configuration Settings That Affect Agents

Element	Description	Default Value
SmtpCipherList	Specifies the list of ciphers that match the cipher suite name that the SMTP server supports. For example, RSA+RC4+SHA. For more information, see 'Advanced SSL Configuration Options' in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.	NA
UseStartTLS	Ignored unless UseSSL is true. If UseStartTls is true, then use the STARTTLS option (RFC 2487) for the SMTP session. Initial connection is through an unsecured link, usually port 25. The connection is then promoted to a secure link using the STARTTLS SMTP command. If UseStartTls is false, then a secured connection is created immediately, before the SMTP protocol is started. This is also known as SMTPS. SMTPS typically uses port 465.	true

20.3.3.3 Agent Scheduler Configuration Settings

Agents are functionally a combination of data that is stored in Oracle BI Presentation Services and Oracle BI Scheduler. The elements in the Scheduler instanceconfig.xml file describe the behavior of all agents that run on a specific Oracle BI Scheduler. Table 20–4 describes each agent configuration element.

Table 20–4 Agent Scheduler Configuration Settings

Element	Description	Default Value
Log_Dir	Agents can create log files if exceptional error conditions occur. Log_Dir specifies the directory where these files are saved. The directory must be accessible to the Oracle BI Scheduler server. In Windows, the default installation runs the service as a system account, which prevents Oracle BI Scheduler from writing to or reading from network directories. If you put script files on network shares, or your scripts access network shares, then Oracle BI Scheduler must be run as a network user. For example: For information on log files, see Section 8.4.5, "Diagnosing Issues with the Agents."	ibots
LogPurgeDays	Specifies the number of days after which old agent logs are deleted automatically. To prevent old logs from being deleted automatically, set the value to 0 (zero).	7
NumGlobalRetries	A Web or mail server that has too many people logged on might reject new connections, including connections from Oracle BI Scheduler. To cope with such overload, an agent retries the connection. This element sets the maximum number of retries to obtain global information about what to deliver and to whom before the agent gives up. If you set this value to 0 (zero), then no retries are attempted.	2
MinGlobalSleepSecs	Specifies the minimum number of seconds that the agent randomly sleeps after its connection is refused before it attempts to reconnect to the server to obtain global information about what to deliver and to whom.	3
MaxGlobalSleepSecs	Specifies the maximum number of seconds that the agent randomly sleeps after its connection is refused before it attempts to reconnect to the server to obtain global information about what to deliver and to whom.	10
NumRequestRetries	After an agent has received the global information, it issues a series of unique requests to the server for each user. This element specifies the number of times Oracle BI Scheduler retries its attempts to connect to the server to issue these requests. If you set this value to 0 (zero), then no retries are attempted.	3
MinRequestSleepSecs	Specifies the minimum number of seconds that the agent randomly sleeps after its connection is refused before it attempts to reconnect to the server to issue requests.	2
MaxRequestSleepSecs	Specifies the maximum number of seconds that the agent randomly sleeps after its connection is refused before it attempts to reconnect to the server to issue requests.	10
NumDelivery Retries	After a unique request has executed, the agent tries to deliver the results to specified devices. This specifies the number of times that Oracle BI Scheduler attempts to retry to connect to the server to deliver the results. If you set this value to 0 (zero), then no retries are attempted.	4
MinDeliverySleepSecs	Specifies the minimum number of seconds that the agent randomly sleeps after its connection is refused before it attempts to reconnect to the server to deliver results.	5
MaxDeliverySleepSecs	Specifies the maximum number of seconds that the agent randomly sleeps after its connection is refused before it attempts to reconnect to the server to deliver results.	10

Table 20–4 (Cont.) Agent Scheduler Configuration Settings

Element	Description	Default Value
MaxRowsTimesColumns	When agents are chained, this value governs the size of filters passed between agents. When you pass a filter to another agent in a chain, Oracle BI Scheduler creates a union of the result sets for the Conditional Report for each personalized recipient. This report can grow very large in some cases (1000 users with 100 unique rows each with ten columns per report = 1,000,000 column values in the filter). The Oracle Business Intelligence servers might not be able to handle such a large filter, so this element specifies the maximum number of rows*columns in the filter.	10,000
Debug	Debug Enabled. Set this element to have Oracle BI Scheduler generate a log file for each agent. This log file has useful logging messages when trying to diagnose a problem. This log file is stored in *ORACLE_INSTANCE*\diagnostics\logs\OracleBISchedulerComponent\coreapplication_obisch*n*. A new log file named Agent-<*Job number*>-<*Instance number*>.log is created for each job instance. The Job Manager can also be used to override the Debug setting for an individual job. For more information, see Section 8.4.5, "Diagnosing Issues with the Agents."	false
KeepErrorLogFiles	Set this element to true to generate an error log file for each agent. This log file is created only when an agent execution encounters errors and contains only error messages. The file is stored in *ORACLE_INSTANCE*\diagnostics\logs\OracleBISchedulerComponent\coreapplication_obisch*n*.	true

20.3.4 Controlling Delivery Options for Agents

Delivery options (that is, delivery devices and delivery profiles) determine how the contents of agents are delivered to users. Delivery options can be configured by users, in the LDAP server (for e-mail addresses), or in the SA System subject area. See Appendix D, "Setting Up the SA System Subject Area" for information.

20.4 Managing Device Types for Agents

You can use different device types from such categories as mobile phones and pagers to deliver the content of agents to users. You can create, view, edit, and delete device types for a device category. Many device types are provided automatically. You can add types that are required for your users.

> **Note:** You can only view system-seeded device types (such as AT&T Wireless); you cannot edit or delete them.

The capability to manage device types is available to users who have the Manage Device Types privilege. For information on privileges, see Section C.2.3.2, "Setting Privileges in Oracle BI Presentation Services Administration."

To create a device type:

1. Log in to Oracle Business Intelligence.

2. In the global header, click **Administration**.
3. Click the **Manage Device Types** link to display the Manage Device Types page.
4. Click the **Create New Device Type** link.
5. Complete the Create New Device Type dialog, and click **OK**.
6. Click **Create Device Type** to return to the Manage Device Types screen.

To view or edit a device type:

1. In the global header, click **Administration**.
2. Click the **Manage Device Types** link.
3. Click the **Edit** button for the appropriate device type.
4. Complete the Edit Device Type dialog, and click **OK**.

To delete a device type:

1. In the global header, click **Administration**.
2. Click the **Manage Device Types** link.
3. Click the **Delete** button for the device type to delete.

   A confirmation box is displayed.
4. Click **OK** to confirm the deletion.

For more information about the Administration page in Oracle Business Intelligence, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

20.5 Monitoring Active Agent Sessions

Using the Manage Agent Sessions page in Oracle BI Presentation Services Administration, you monitor currently active agent sessions that are triggered by Oracle BI Scheduler. For example, you can see a list of active agents per session.

When one or more agent sessions are active, information about each agent session is displayed, such as the job identifier and the instance identifier that are assigned to the agent session by the Oracle BI Scheduler. Expanding the agent session shows the individual agents (one agent, or multiple agents if they are chained). The state of the agent is either Created, Populated, or Conditional Request Resolved.

Expanding a specific agent in a particular session shows the recipients for the agent and their type, such as the Engineering recipients defined in a group, or individual users. When the recipient is a group, the individual members of the group are not listed.

Note: When agents are chained, the recipient list is depends on the parent agent. The recipients are shown for the parent agent definition only, and not for the actual execution of chained agents.

To view information about active agent sessions:

1. In the global header, click **Administration**.
2. Click the **Manage Agent Sessions** link to display the Manage Agent Sessions page and do one of the following:

- To sort agent sessions by their values in a particular column, click the **Sort** button for that column.

  Re-sorting the list causes the page to refresh so the number of active agent sessions might increase or decrease as a result.
- To view more information about an agent session or about agents within a particular session, click the **Expand** button.
- To view the definition of an individual agent, click its link.

For more information about the Administration page in Oracle Business Intelligence, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

21

Configuring Advanced Options for Mapping and Spatial Information

This chapter describes advanced configuration options that you can set for map views. It includes the following topics:

- Section 21.1, "Configuring MapViewer to Support Map Views"
- Section 21.2, "Manually Configuring for Map Views"
- Section 21.3, "Inserting Text on a Map"
- Section 21.4, "Embedding Maps in External Pages"

Before configuring for map views, ensure that you are familiar with the information in the following guides:

- *Oracle Fusion Middleware User's Guide for Oracle MapViewer*, which is part of the Oracle Fusion Middleware documentation library.
- *Oracle Spatial Developer's Guide*, which is part of the Oracle Database documentation library.

21.1 Configuring MapViewer to Support Map Views

MapViewer is installed as part of Oracle BI Enterprise Edition and deployed in the same domain as Oracle BI EE on the web application server. The default context path of MapViewer in the application server is /mapviewer. You can use the administration console in MapViewer to configure it for use with map views.

You can configure a separate remote instance just for MapViewer to act as a proxy that supports the heavy processing load that maps require. If performance is not a major concern, then you can use a MapViewer instance that is co-located with Oracle BI EE as the rendering engine.

The MapViewer engine can serve in the following roles:

- Co-located MapViewer — Also known as non-proxy mode. If the MapViewer is located in the same domain as Oracle BI EE and used as the rendering engine, then all map resources (such as JavaScript files and images) are downloaded from that instance of MapViewer.
- Remote MapViewer — Also known as proxy mode. If a separate remote instance of MapViewer is configured as the rendering engine, then the browser cannot communicate with the remote instance for resources. Browsers do not allow cross-domain AJAX calls for security reasons. To overcome this limitation, all requests are first forwarded to the co-located MapViewer, which in turn communicates with the actual remote instance.

Complete the following steps to configure for a remote MapViewer:

- Edit the RemoteOracleMapViewerAbsoluteURL element in the instanceconfig.xml file, as described in Section 21.2, "Manually Configuring for Map Views."
- Edit the proxy_enabled_hosts element in the mapViewerConfig.xml configuration file for MapViewer to point to the MapViewer on the remote server, as shown in the following example:

```
  <security_config>
    <proxy_enabled_hosts>http://remoteserver:9704/mapviewer</proxy_enabled_
hosts>
  </security_config>
```

21.2 Manually Configuring for Map Views

Use various elements in the instanceconfig.xml file to configure map views. Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings for configuring map views:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Search for the SpatialMaps section, in which you must add the following elements:
   - ColocatedOracleMapViewerContextPath — Specifies the context path of the MapViewer instance that is co-located with Oracle BI EE. The default value is /mapviewer.
   - RemoteOracleMapViewerAbsoluteURL — Specifies the URL of the remote MapViewer instance. This element has no default value.

     If this element has no value, then the system assumes that the map rendering engine is the co-located MapViewer instance (such as /mapviewer). If this element has a value, then the co-located MapViewer acts as proxy for all requests for the remote server. The following example shows a sample value:

     ```
     <RemoteOracleMapViewerAbsoluteURL>http://remoteserver:9704/mapviewer
     </RemoteOracleMapViewerAbsoluteURL>
     ```
   - MaxRecords — Specifies the maximum number of records that can be included in a layer on the map. The setting applies to all layers on the map and overrides the MaxVisibleRows element that applies to data cubes. The default value is 500. If the format for a layer causes this value to be exceeded, then a warning message is displayed. The parent element is LayerDataLayout.
   - SyndicatedOracleMapViewerContextPath — Specifies the URL of the MapViewer instance for embedding maps in external pages. For details and an example, see Section 21.4, "Embedding Maps in External Pages."
3. Include the elements and their ancestor elements as appropriate, as shown in the following example.

```
<ServerInstance>
   <SpatialMaps>
      <ColocatedOracleMapViewerContextPath>/mapviewer</ColocatedOracleMapViewer
ContextPath>
      <RemoteOracleMapViewerAbsoluteURL></RemoteOracleMapViewerAbsoluteURL>
      <LayerDataLayout>
        <MaxRecords>600</MaxRecords>
```

```
        </LayerDataLayout>
    </SpatialMaps>
</ServerInstance>
```

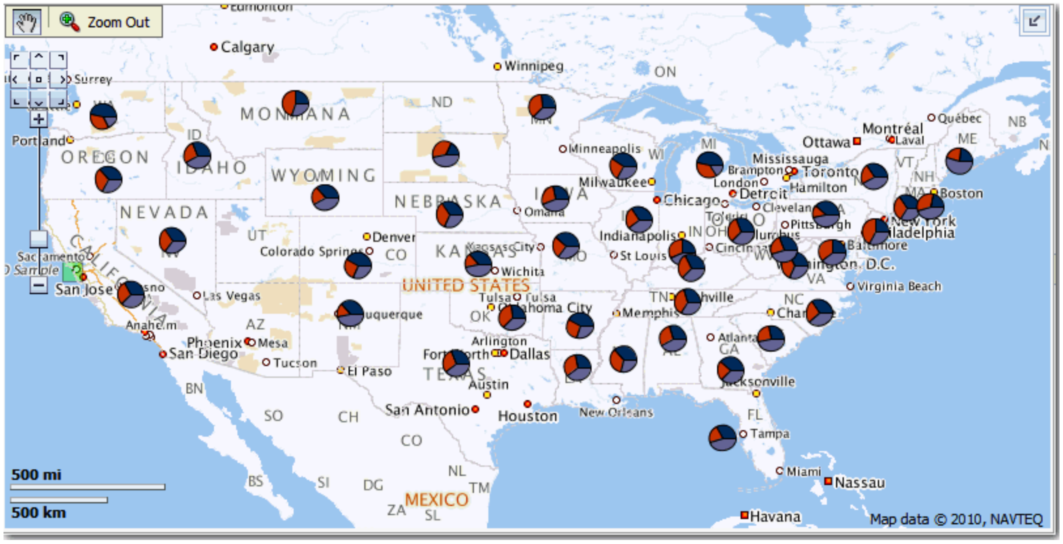
4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

21.3 Inserting Text on a Map

You can add any text, such as a copyright string, to the tile layer definition of a map. The text is automatically updated on the map in Oracle BI EE when a tile layer is added or deleted or becomes invisible. The position of the text is also automatically adjusted when the user minimizes, restores, or removes the overview map.

Figure 21–1 shows an example of a copyright string on a map. The string is in the lower-right corner.

Figure 21–1 Copyright String on a Map



To insert text in the tile layer definition on a map:

1. Create the tile layer.

   For information, see *Oracle Fusion Middleware User's Guide for Oracle MapViewer*.

2. Edit the tile layer definition by selecting it and clicking the **Edit / View Details** button.
3. Click the **XML Mode** button.
4. In the XML code, enter the <copyright> tag immediately after the <map_tile_layer name> tag.
5. Enter the appropriate text within the <copyright> tag.

   For example, the following code displays the copyright line that is shown in Figure 21–1.

   <copyright>Map data © 2010, NAVTEQ</copyright>

6. Click the **Submit** button to save your changes.

7. If you do not see the updated text on the map, then purge the cached geometries.

   For information, see *Oracle Fusion Middleware User's Guide for Oracle MapViewer*.

21.4 Embedding Maps in External Pages

You can embed map views in external pages, such as those from Oracle Portal. To embed views that are hosted in a separate Web application server, you can follow the proxy rules that are outlined in the previous sections. Because of browser restrictions, install the MapViewer instance in the same application server as external pages or portals. The proxy context path of the MapViewer instance that is installed in the Web application server that hosts external pages can differ from the application server that is hosting content for Oracle BI EE. In this case, set the SyndicatedOracleMapViewerContextPath element. When the server for Oracle BI Presentation Services identifies a request that originated from a third-party page, the server checks the element value to determine where to pass the proxy requests.

The following example provides sample values for this element.

```
<ServerInstance>
   <SpatialMaps>
      <SyndicatedOracleMapViewerContextPath>/mapviewer</SyndicatedOracleMapViewerC
ontextPath>
      <RemoteOracleMapViewerAbsoluteURL>http://myserver:9704/mapviewer</RemoteOrac
leMapViewerAbsoluteURL>
   </SpatialMaps>
</ServerInstance>
```

22

Customizing the Oracle BI Web User Interface

This chapter describes how to customize the appearance of the Oracle BI Presentation Services user interface in a Web browser.

This chapter contains the following sections:

22.1 Tips for Customizing the Web User Interface

As you plan to customize the web user interface, keep the following points in mind:

- When customizing, you often copy files that are installed with Oracle Business Intelligence into a directory in which you can modify them. By modifying files in this directory, you prevent your modifications from being overwritten when the software is upgraded.

  For performance reasons, copy only those files that you plan to modify, due to the lookup process for files. Oracle BI EE first looks for files in the directories that have been specified for your modifications, then it looks in the directories in which it installed standard files.

- In Oracle BI Presentation Services, customization of user interface elements and appearance is accomplished by modifying XML message files, and styles and skins, and not with JavaScript. You should not modify JavaScript files that are located in the *ORACLE_HOME*\bifoundation\web\app\res directory. This is because the objects and methods in these scripts might change, and because these files might be replaced when upgrading.

  (In a dashboard, users with the appropriate permissions can customize an individual dashboard section by adding HTML to it. This HTML can include JavaScript. For more information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.)

- You can adapt the Oracle BI EE deployment to a particular language. For information, see Chapter 16, "Localizing Oracle Business Intelligence".

22.2 What are Skins and Styles?

You can control the way that the interface for Oracle BI EE is displayed to users by creating skins and styles. The primary difference between skins and styles is that skins can be automatically assigned to a user at login time, while users can select styles at run time to control things such as table formatting, and so on. The default skins and styles are located in the application resources directory (*ORACLE_HOME*\bifoundation\web\app\res) in the sk_* and s_* directories.

A "skin" is assigned based on the value of the SKIN system session variable. The user can then alter certain elements, such as fonts, colors, various attributes of tables and graphs, and so on, by picking a "style" for analysis results (or a dashboard) when they are logged into Oracle BI EE. Skins consist of non-alterable elements, such as corporate logos or other graphics.

A "style" controls how dashboards and results are formatted for display, such as the color of text and links, the font and size of text, the borders in tables, the colors and attributes of graphs, and so on. Styles consist of alterable elements, often combined into style sheets.

While skins and styles are typically used to customize the look and feel of analyses and dashboards by providing logos, color schemes, fonts, table borders, and other elements, they can also be used to control the position and justification of various elements by including specialized style tags in the relevant style sheet (.css) file.

22.2.1 Using the SKIN Variable

The SKIN variable points to a directory that contains non-alterable elements. Such directories are located in the *ORACLE_HOME*\bifoundation\web\app\res directory and begin with "sk_". For example, if a directory were called sk_MyCompany, the SKIN variable would be set to MyCompany. Styles are organized into similar directories that contain the alterable elements. Style folders and style sheets are located in the resources directory in directories that begin with "s_". For example, if a directory were called s_MyCompanyStyle, then the style name would be My CompanyStyle.

Both skin and style directories can contain Cascading Style Sheets (files with a .css extension), images, and graph templates. You can create new styles and skins by creating new style and new skin directories.

Note: Style and skin names cannot include underscores.

22.3 Modifying Oracle BI Presentation Services User Interface Styles

You can modify the Cascading Style Sheets (CSS) files and the images that are stored in the default installation directory to create a custom user interface. The default images and style sheets are located in the *ORACLE_HOME*\bifoundation\web\app\res directory. The relevant subdirectories are contained in this directory for the current Oracle Business Intelligence style.

Cascading style sheets provide control over any object within Oracle Business Intelligence. You can change images, backgrounds, font colors and sizes, table cell gridlines and cell padding, and so on. For more information about cascading style sheets, consult a resource such as the Microsoft Developer Network (MSDN).

This section contains the following topics:

- Section 22.3.1, "Creating a New Dashboard Style"

- Section 22.3.2, "Specifying Defaults for Styles and Skins"

22.3.1 Creating a New Dashboard Style

The easiest way to create a style is to copy the *ORACLE_HOME*\bifoundation\web\app\res\s_blafp and \sk_blafp directories and paste them into the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\res directory. Copying to this directory rather than to the main installation directory prevents any customized CSS files and images from being overwritten during a software upgrade.

The b_mozilla directory contains the important files for making quick changes to the dashboards.

To create a style:

1. Copy the s_blafp and sk_blafp subdirectories of the *ORACLE_HOME*\bifoundation\web\app\res directory.
2. Paste them to the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\res directory.
3. Make and save your modifications.
4. Restart the service for Oracle BI Presentation Services to display the style sheet name in the Dashboard Properties dialog.

22.3.2 Specifying Defaults for Styles and Skins

You can specify which style and skin to use when users choose the default style in the Dashboard Properties dialog, using the DefaultStyle and DefaultSkin elements in the instanceconfig.xml file.

You can specify which style folder to use in the *ORACLE_HOME*\bifoundation\web\app\res directory when users select the Default option from the Styles list in the Dashboard Properties dialog. If the style folder begins with the characters s_, such as s_TestStyle, then omit those characters when specifying the value for the DefaultStyle element.

To specify the skins folder that is paired with the style folder, use the DefaultSkin element. If the skins folder begins with the characters sk_, such as sk_TestSkin, then omit those characters from the value.

If users do not make a choice or if these entries are not present in the instanceconfig.xml file, then the styles and skins that are installed with Oracle BI EE are used. These styles and skins are located in the *ORACLE_HOME*\bifoundation\web\app\res directory. The default value for both elements is blafp.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings that controls the default styles and skins:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the DefaultStyle and DefaultSkin elements.

3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

   ```
   <ServerInstance>
       <DefaultStyle>myStyle</DefaultStyle>
       <DefaultSkin>mySkin</DefaultSkin>
   </ServerInstance>
   ```

4. Save your changes and close the file.
5. Restart Presentation Services.

22.4 Customizing Language Selections and Other Components

You can customize various aspects of the user interface, including the following ones:

- Section 22.4.1, "Customizing a Language Selection"
- Section 22.4.2, "Customizing the Appearance of the Sign-In Page"
- Section 22.4.3, "Customizing Non-Dashboard Components"

22.4.1 Customizing a Language Selection

The Oracle Business Intelligence sign-in page and the My Account dialog each include a language selection list, where users select the language in which they want to work. If a language that users want to select is not displayed in the list, then you can add it.

To add a language to the selection list on the sign-in page and the My Account dialog:

1. Add the l_*xx* directory to the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\messages directory, where *xx* is the language extension for the language to add (for example, en for english-usa).
2. Copy the languagenames.xml file in the *ORACLE_HOME*\bifoundation\web\msgdb\messages directory to the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\messages directory. (The languagenames.xml file contains the list of languages that are acceptable for the language selection lists.
3. Use a text editor to open the languagenames.xml file in the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\messages directory.
4. Ensure that the language to add is not listed in the file. If it is not listed, then make the following entry to add the language:

   ```
   <WebMessage name="kmsgLanguageName_xx">
   <TEXT>LanguageName</TEXT>
   </WebMessage>
   ```

   where *xx* is the language extension (for example, ar) and *LanguageName* is the name of the language (for example, Arabic).

For more information on languages and localization, see Chapter 16, "Localizing Oracle Business Intelligence".

22.4.2 Customizing the Appearance of the Sign-In Page

Users first enter the appropriate URL in a Web browser, then they see the sign-in page for Oracle Business Intelligence. Users must sign in to Oracle Business Intelligence with an appropriate user name and password to gain access to the various editors and components. When sign-in is successful, users generally see their default dashboard.

You can customize the sign-in page to suit the needs of your organization. You cannot control how users are actually authenticated. For information about authentication options, see Appendix C, "Managing Security for Dashboards and Analyses."

To customize the appearance of the sign-in page:

1. Copy the logoncontrolmessages.xml file in the *ORACLE_HOME*\bifoundation\web\msgdb\messages directory to the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\messages directory.
2. Edit the logoncontrolmessages.xml file as appropriate. Save your changes and close the file.
3. Copy the files to modify from the *ORACLE_HOME*\bifoundation\web\msgdb\pages\common directory to the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\pages\common directory.
4. Edit the files in the \pages\common directory as appropriate, keeping the following points in mind:
   - The files all end with the .html suffix. However, these are actually XML-based files. When editing the files, use an editor that supports XML and ensure that the resulting files are saved as valid XML.
   - The Not Logged On page (kmsgAuthenticateNotLoggedOn), used for session timeout, is displayed only when users are not logged on and attempt to access a URL that does not support direct logon. For example, suppose a user accesses Oracle BI Answers and clicks the **Log Off** link. If the user clicks the browser's **Back** button and then clicks the **My Account** link, then the user receives the Not Logged On page.
5. Save your changes and close the file.

For general information about customizing the user interfaces, see Section 22.5, "Customizing the User Interface Using XML Message Files".

22.4.3 Customizing Non-Dashboard Components

Non-dashboard components include the Analysis editor, the agent editor, and the Oracle BI Presentation Services Administration page. The CSS files for these components are stored in the sk_ directory within the main installation directory (*ORACLE_HOME*\bifoundation\web\app\analyticsRes). The b_mozilla_4 directory contains the relevant CSS files that correspond to these components.

Use the same logic that is described in Section 22.3, "Modifying Oracle BI Presentation Services User Interface Styles" to make modifications to the non-dashboard components of the user interface.

The non-dashboard components are controlled globally. Users cannot toggle between multiple user interfaces for the non-dashboard components.

22.5 Customizing the User Interface Using XML Message Files

This section explains how to customize text elements in message files to manage the default appearance and behavior of the Oracle BI Presentation Services user interface using XML strings.

> **Note:** The intent of this section is to allow organizations that have XML expertise to perform additional customization. If you do not have this expertise, then enlist the assistance of a third party to help you with customization.

This section contains the following topics:

- Section 22.5.1, "User Interface XML Message Files"
- Section 22.5.2, "Structure of XML Message Files"
- Section 22.5.3, "Customizing XML Messages"
- Section 22.5.4, "Resolution of XML Message Name Elements"
- Section 22.5.5, "Sample XML Template"
- Section 22.5.6, "Sample custommessages.xml File"

> **Note:** Other topics in this guide describe additional customizations that you can perform by customizing text elements in message files, such as Section 19.7, "Specifying View Defaults for Analyses and Dashboards."

22.5.1 User Interface XML Message Files

You can customize many of the text elements that are displayed in the Analysis editor, the agents editor, and on dashboard pages. Examples of text elements include the content of text strings, the text for prompts such as the names of links and buttons, and the text of error and informational messages that are displayed to users as the result of their work.

These text elements are contained in external message files that are installed with Oracle Business Intelligence. The message files are in XML format. Language-specific messages are located in the *ORACLE_HOME*\bifoundation\web\msgdb\l_*xx*\messages directory, where *xx* is the language identifier of the selected locale (for example, for english-usa, the identifier is en). Language-independent messages are located in the *ORACLE_HOME*\bifoundation\web\msgdb\messages directory.

You should not edit the message files directly because any changes would not be retained when you install newer versions or service releases. For more information, see Section 22.5.3, "Customizing XML Messages."

22.5.2 Structure of XML Message Files

The name of a particular message file indicates the kind of content that it holds. For example, messages in the logonmessages.xml file hold message content that relates to the act of signing in and out of the application. Within each XML file, the WebMessage name= elements define the names of the messages. These elements are called *message identifiers*.

A particular message might also reference the content of another message by using a MessageRef element. For example, the following message in the logonmessages.xml file references the value of another message:

```
<WebMessage name="kmsgAuthenticateNotLoggedOnToLogOnClickHere">
 <HTML>
  You are not currently logged in to the
  <MessageRef name="kmsgProductServer" />
```

The entry `<MessageRef name="kmsgProductServer" />` in the previous message indicates that the name of the server is taken from the value of the kmsgProductServer message identifier. This message is located in the productmessages.xml file, and its value is Oracle BI Server:

```
<WebMessage name="kmsgProductServer" CRC="nnnnnnnnnnnnnnnnnnnn">
 <TEXT>Oracle BI Server</TEXT>
```

Some messages, such as those that contain copyright information and product names, are protected and cannot be changed. The productmessages.xml file contains text preceding the WebMessage elements that indicates that the associated names cannot be changed.

22.5.3 Customizing XML Messages

This section explains how to change the content of unprotected messages and provides several examples. The intent is to show you how to alter the text of messages, and not to teach you XML.

To customize messages:

1. Create message identifiers with similar names and customize their text.
2. Create a custom messages directory named customMessages.

   > **Note:** Organizations that have Oracle Business Intelligence applications might have a file present in this directory. This file enables Oracle Business Intelligence support for Oracle Business Intelligence applications, and should not be modified, moved, or deleted.

3. Place the messages in one or more XML files in the customMessages directory, and then place the customMessages directory in this location:

   ORACLE_INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*

   where *xx* is the language identifier of the selected locale (for example, for english-usa, the identifier is en).

   If you are not concerned with multiple languages, then place the directory in the l_en directory. Messages default to l_en if a language-specific version is not found. You must create the l_*xx* directory in the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb directory.

4. Restart Oracle BI Presentation Services.

You can create multiple XML files in the customMessages directory, or you can create a single XML file that holds customized messages, for example, custommessages.xml.

This is because the application accesses the customMessages directory and reads every file that has an XML extension, regardless of the file's name. If you create many customized messages, then you might prefer to organize them into separate files.

> **Note:** If you intend to support multiple languages, then place control messages (which are messages that are not translated) into one file named customcontrolmessages.xml. Place messages that are translated into another file named, for example, customuimessages.xml. This places localized versions of the customuimessages.xml file in each language directory as appropriate, such as *ORACLE_HOME*\bifoundation\web\msgdb\l_de\customMessages, *ORACLE_HOME*\bifoundation\web\msgdb\l_fr\customMessages, and so on.

Links are a special case. Modifications made to link text are displayed as expected in dashboards and the agents editor. To make these same modifications be displayed in the Analysis editor, you must modify the kuiAnswersMainBar message.

To edit a custom message file:

1. Make a backup of the original file in a separate directory.
2. Make a development copy in a different directory.
3. Edit the development version of the file in a text or XML editor.
4. Replace the original file in the customMessages directory with the newly edited file.
5. Test the new file.
6. (Optional) Delete the backup and development copies.

22.5.4 Resolution of XML Message Name Elements

During initialization, WebMessage name default text is replaced with text from equivalently named elements in any customized XML file, based on the following precedence order, from highest to lowest:

- XML in *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*\customMessages directory, which are language-specific directories.
- XML in *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_en\customMessages directory, for language-specific user sign-ins if WebMessage name elements reside here, but are not in language-specific files.
- XML in *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\customMessages directory.
- XML in *ORACLE_HOME*\bifoundation\web\msgdb\l_*xx*\messages directory.
- XML in *ORACLE_HOME*\bifoundation\web\msgdb\messages directory.

As an example, when Oracle BI Presentation Services starts, it first reads the messages in the *ORACLE_HOME*\bifoundation\web\msgdb\l_*xx*\messages directory, and then reads the messages in the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*\customMessages directory. It replaces the default text for those messages with the customized text. If you attempt to alter the text of a protected message, then a message is displayed in its place that indicates that you attempted this.

22.5.5 Sample XML Template

The following is a sample template for a custommessages.xml file in the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*\customMessages directory. An example custommessages.xml file follows the template.

Every message begins with a <WebMessage name=> tag and ends with a </WebMessage> tag. The message text that you can customize is contained with the TEXT elements or HTML element. To suppress the display, delete the text between the start and end tags.

```
    <?xml version="1.0" encoding="utf-8"?>
    <WebMessageTables>
         <WebMessageTable system="Custom Messages">
         <!-- The name of a message must match the name of the message you are
overriding. -->
         <WebMessage name="kmsgExampleOverrideMessage">
         <!-- A message can have TEXT and HTML versions of it. It is not necessary
to have both. (TEXT is automatically converted to HTML when necessary). -->
         <TEXT>Example message.</TEXT> <!-- Format used in a text only output -->
         <HTML><b>Example message with bold HTML tags.</b></HTML> <!-- Format used
in an HTML output -->
    </WebMessage>
</WebMessageTable>
</WebMessageTables>
```

To create a sample template:

1. Replicate the sample template in a text editor.
2. Name the file custommessages.xml (or any name you choose that ends with .xml).
3. Place the file in the customMessages directory that you created in the appropriate *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx* directory.

22.5.6 Sample custommessages.xml File

The following example shows four customized messages placed in the custommessages.xml file.

```
    <?xml version="1.0" encoding="utf-8"?>
    <WebMessageTables>
   <WebMessageTable system="Custom Messages">
    <!-- First message -->
         <WebMessage name="kmsgAuthenticateRemembermyIDandpassword">
         <TEXT>Remember my sign-in name and password.</TEXT>
    </WebMessage>
    <!-- Second message -->
         <WebMessage name="kkmsgPrivilegeDisplayerAccountUnknown">
         <TEXT>Unknown Account (<Param insert="1"/>). Call the Help Desk at
```

```
extension 9999 to set up a new account.</TEXT>
    </WebMessage>
    <!-- Third message --
         <WebMessage name="kmsgWelcomeFrameCreateNewRequest">
         <HTML>Create a <b>new analysis</b> by clicking on a Subject Area below.
After creating the analysis, click the <b>Done</b> button at the bottom of the
page.</HTML>
    </WebMessage>
    <!-- Fourth message -->
         <WebMessage name="kmsgUIAdmin">
         <HTML></HTML>
    </WebMessage>
</WebMessageTable>
</WebMessageTables>
```

- The message identifier of the first message being customized is "kmsgAuthenticateRemembermyIDandpassword". The default text for this message is located in the file logonmessages.xml in the *ORACLE_HOME*\bifoundation\web\msgdb\l_*xx*\messages directory.
- The message identifier of the second message being customized is "kmsgPrivilegeDisplayerAccountUnknown". The default text for this message is located in the file viewmessages.xml in the *ORACLE_HOME*\bifoundation\web\msgdb\l_*xx*\messages directory. This message contains a variable,

  ```
  ( <Param insert="1"/> ).
  ```

> **Note:** If you are customizing a message that contains a variable, then do not alter the variable. In the UNIX environment, be careful to preserve the case of the message name being customized.

- The message identifier for the third message being customized is "kmsgWelcomeFrameCreateNewRequest". The default text for this message is located in the file searchsysmessages.xml in the *ORACLE_HOME*\bifoundation\web\msgdb\l_*xx*\messages directory. This message is in HTML format and uses an HTML tag (<b>) to display text in bold letters.
- The message identifier for the fourth message being customized is "kmsgUIADMIN". The default text for this message is located in the file uimessages.xml in the *ORACLE_HOME*\bifoundation\web\msgdb\l_*xx*\messages directory. This message is in HTML format. This message identifier displays the Administration link at the top of each dashboard page or tab in the Analysis editor or agent editor. Deleting the Administration text between the <HTML> and </HTML> tags suppresses the display of the link.

23

Configuring Resource Availability and URL Generation

You can configure how resources are made available for HTTP access and how URLs are generated by Oracle BI Presentation Services. To do so, you modify the instanceconfig.xml file to include the URL element and its interrelated sub-elements, as described in the following procedure.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings for resource availability and URL generation:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the elements that are described in Table 23–1.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <URL>
    <AbsoluteCommandURLPrefix>value</AbsoluteCommandURLPrefix>
    <CustomerResourcePhysicalPath>value</CustomerResourcePhysicalPath>
    <CustomerResourceVirtualPath>value</CustomerResourceVirtualPath>
    <ForceAbsoluteResourceURL>value</ForceAbsoluteResourceURL>
    <ResourcePhysicalPath>value</ResourcePhysicalPath>
    <ResourceServerPrefix>value</ResourceServerPrefix>
    <ResourceVirtualPath>value</ResourceVirtualPath>
  </URL>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

Table 23–1 Elements for URL Generation and Resource File Location

Element	Description	Default Value
AbsoluteCommandURLPrefix	Specifies how Presentation Services generates command URLs. If you explicitly specify an value, then it must be of the following form: `protocol://server/virtualpath` where *virtualpath* is the complete virtual path to Presentation Services. The default is determined separately for each client, based on the URL that the client sends to Presentation Services.	Varies
CustomerResourcePhysicalPath	Specifies the physical location of resource files that are not part of a default installation. Such resource files include customized styles and skins. The internal default is *ORACLE_HOME*\bifoundation\web\app\res. You must provide a full path. Presentation Services must have read permission to this path. For example, if this is a shared network resource, then you must ensure that the user under which Presentation Services is running has read access to the shared resource and read access to the file system from which the shared resource is exported.	Varies
CustomerResourceVirtualPath	Specifies the virtual path used for resource files that are not part of a default installation as specified in the CustomerResourcePhysicalPath element.	NA
ForceAbsoluteResourceURL	Specifies whether Presentation Services always generates fully qualified URLs for resource files that have fully qualified virtual paths. When set to False, resources and the Presentation Services extension are served from one server. When set to True, default resources are served from the same server as the Presentation Services extension, and customer resources are served from another server. Depending on the value of the other settings described in this table, you can also configure to have default and customer resources served from one server, and the Presentation Services extension served from another server.	False

Table 23–1 (Cont.) Elements for URL Generation and Resource File Location

Element	Description	Default Value
ResourcePhysicalPath	Specifies the physical location of the primary resource files for Presentation Services. These are the resource files that are distributed with Presentation Services, not user-customized files such as custom styles or skins. The internal default is *ORACLE_HOME*\bifoundation\web\app\res. You must provide a full path. Presentation Services must have read permission to this path. For example, if this is a shared network resource, then you must ensure that the user under which Presentation Services is running has read access to the shared resource and read access to the file system from which the shared resource is exported. If the value for this entry is different from the physical location of the DLLs for Presentation Services, then you must specify a value for the ResourceVirtualPath element.	NA
ResourceServerPrefix	Specifies how Presentation Services generates URLs for static resources such as images, script files, style sheets, and other user-specified files. The default is *protocol*://*server* from the AbsoluteCommandURLPrefix element. If you explicitly specify a value, then it must be of this form: `protocol://server` If you specify a virtual path, then it is removed. This element designates a separate Web server for delivering static resources, thereby reducing the load on the main Web server. This prefix is used for the resources that have a fully qualified virtual path of the form '/Path/file'. If a resource file has a relative virtual path of the form 'Path/file', then the prefix used is the same one that is used for commands to the Presentation Services extension.	*protocol*://*server*

Table 23–1 (Cont.) Elements for URL Generation and Resource File Location

Element	Description	Default Value
ResourceVirtualPath	Specifies the virtual path used for the primary resource files for Presentation Services, as specified by the ResourcePhysicalPath element. These resource files and customer-defined resource files must be served from the same Web server. For generating relative URLs, the virtual path defaults to res, if the resource folder is present under the same virtual directory as the Oracle BI Presentation Services DLL files. For generating absolute URLs, the value of the AbsoluteCommandURLPrefix element is used as the default. The value must be a fully qualified virtual path of this form: `'/`*`VirtualPath`*`'` If you omit the leading slash, then one is added.	res

Part VII

Configuring Oracle BI Scheduler

This part explains how to use the Oracle BI Scheduler. It contains the following chapters:

- Chapter 24, "Introducing Oracle BI Scheduler"
- Chapter 25, "Configuration Tasks for Oracle BI Scheduler"
- Chapter 26, "Using Oracle BI Scheduler Job Manager"
- Chapter 27, "Programming BI Scheduler VBScript and JScript Jobs"
- Chapter 28, "Programming BI Scheduler Java Jobs"
- Chapter 29, "Oracle BI Scheduler Job Manager Menus"

24

Introducing Oracle BI Scheduler

This chapter provides general information about Oracle BI Scheduler. It contains the following topics:

24.1 What is Oracle BI Scheduler?

Oracle BI Scheduler is a server that manages and schedules jobs. Oracle BI Scheduler supports two kinds of jobs:

- Scripted jobs.

  Scripted jobs are configured and submitted using the Job Manager feature of the Oracle BI Administration Tool. For example, a scripted job could periodically load the Oracle BI Server usage statistics into a back-end database. In this example Oracle BI Scheduler communicates with the BI Server. However, scripted jobs might not access the BI Server, for example by saving the output of an agent to a shared drive. Scripted jobs can also be configured through agents and actions.

  Oracle BI Scheduler supports jobs that are written in the Java programming language or in the VBScript and JScript scripting languages.

  > **Note:** Scripting for agents and scripts that are defined by the Oracle BI Scheduler Job Manager are supported only under Windows platforms. The Java interfaces support all platforms.

- Agents.

  Agents deliver content to end users. Content can be analyses, dashboards, briefing books, or alerts. After delivering content, agents can also execute actions. Actions include Java actions, URL actions, Web service actions, and server script actions. Agents can also run other agents, creating chains of agents. Agents are configured and submitted for execution using Oracle BI Delivers. Oracle BI Scheduler communicates with Oracle BI Presentation Services for unscripted jobs.

> **Note:** There are thus two different forms of custom Java that can be executed: Java scripted jobs run in their own right and Java actions run as part of an agent.

24.2 About Oracle BI Scheduler Server Components

Oracle BI Scheduler consists of the following components:

- Oracle BI Scheduler Job Manager
- Oracle BI Scheduler Service process:
  - Windows operating systems: `nqscheduler.exe`
  - UNIX operating systems: `nqscheduler`
- Scheduler tab in Fusion Middleware Control
- Command line job invocation tool:
  - Windows operating systems: `saschinvoke.exe`
  - UNIX operating systems: `saschinvoke`

24.3 About Agent Support in Oracle BI Scheduler

When a user creates and schedules an agent, Oracle BI Presentation Services gathers information about the agent such as its priority, the intended recipients, and the devices to which content should be delivered. Presentation Services packages this information and other characteristics into a job, then informs Oracle BI Scheduler when to execute the job.

Agents can run in parallel on different threads. The number of agents that can run in parallel depends on the size of the Scheduler thread pool (a configurable setting) and the number of threads used up by each agent. Queueing might occur if too many agents are triggered at the same time.

For information about agents, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

24.4 About Database Support in Oracle BI Scheduler

Oracle BI Scheduler uses a single back-end database to store pertinent information about a job, its instances, and its parameters. For information, see Section 1.8, "System Requirements and Certification".

The Oracle BI Scheduler service starts only if the back-end database satisfies the following requirements:

Back-End Database Requirement	Notes
Configured	There is a one-to-one relationship between the back-end database schema and Oracle BI Scheduler. Do not configure multiple Oracle BI Scheduler applications to use a single database schema.
Operational	For information about the specific back-end databases that are supported by Oracle BI Scheduler, see Section 1.8, "System Requirements and Certification".

Back-End Database Requirement	Notes
Mixed Authentication Mode	For example, if the database is SQL Server, then the security mode should be set to allow both SQL Server and Windows Authentication for logon. Do not use operating system authentication for the back-end database login. If you do, then the Oracle BI Scheduler service might not start in some cases.

24.5 About Oracle BI Scheduler Schemas

You create the database schemas that are associated with Oracle BI Scheduler using the Repository Creation Utility (RCU). (For information, see Section 25.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU.") You can also create tables in Oracle BI Scheduler's schema for use with other major commercial databases, by running the SQL scripts in the following directory:

\orainst\bifoundation\OracleBISchedulerComponent\coreapplication_obisch1\schema\.

> **Note:** To upgrade Scheduler tables, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*.

Table 24–1 gives brief descriptions of the database tables that are used by Oracle BI Scheduler.

If usage tracking is enabled in the NQSConfig.INI file, then the Oracle BI Server generates Usage Tracking data files. A sample JavaScript is provided, which extracts information from the Usage Tracking files and loads them to a table in the relational database. The S_NQ_ACCT table stores all the information regarding Accounting Data. For information, see Chapter 9, "Managing Usage Tracking.".

Table 24–1 Tables Used by Oracle BI Scheduler

Table Name	Table Description
S_NQ_JOB	Stores information about scheduled jobs.
S_NQ_INSTANCE	Stores information about scheduled job instances.
S_NQ_ERR_MSG	Stores error messages about Oracle BI Scheduler job instances that do not complete successfully. Also stores information that coordinates restarting an agent, if there is a Scheduler failover during agent execution.
S_NQ_JOB_PARAM	Stores information about Oracle BI Scheduler job parameters for scheduled jobs. For example, agent path.

The following are characteristics of Oracle BI Scheduler schema tables:

- The contents of the schema tables are implementation-specific. For information about creating these tables, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.
- Schemas store path and file names to job scripts, which provide easy updates to several jobs if they share a script.

25

Configuration Tasks for Oracle BI Scheduler

Depending on your specific deployment, you must perform the following Oracle BI Scheduler configuration tasks:

- Section 25.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU"
- Section 25.2, "Configuring a Database for the Oracle BI Scheduler"
- Section 25.3, "Configuring Oracle BI Scheduler Instances in a Clustered Environment"
- Section 25.4, "Changing Oracle BI Scheduler Table Names"

Keep the following points in mind:

- If you are not using Oracle BI Scheduler, then you do not need the information in this chapter.
- Configuration that is required for running agents (as opposed to running jobs in general) is described in Chapter 20, "Configuring and Managing Agents."
- If you are migrating an Oracle Business Intelligence environment to a new system, then ensure that you also migrate the Oracle Business Intelligence Server repository file and the Oracle BI Scheduler tables. For information, see Chapter 32, "Moving to a New Environment." The Oracle BI Scheduler tables are required for agents.

25.1 Creating an Oracle BI Scheduler Database and Tables Using RCU

You use the Repository Creation Utility (RCU) to create Oracle BI Scheduler database schemas and tables. The RCU is a graphical tool that enables you to create and manage Oracle Fusion Middleware database schemas in your database.

The RCU enables you to use either an existing database schema or to create a database schema in which to store the Oracle BI Scheduler tables.

For information about using the RCU, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

25.2 Configuring a Database for the Oracle BI Scheduler

You can use Fusion Middleware Control to configure common settings that are used by agents with Oracle BI Scheduler.

The following procedures describe how to configure the back-end database and tables:

- Section 25.2.1, "Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler."

- Section 25.2.2, "Configuring a System DSN entry for SQL Server Databases."

25.2.1 Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler

The Data Source Name that is specified in the Fusion Middleware Control configuration must match the database service alias that was created in Section 25.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."

The following procedure configures database connection details. Before you begin this procedure, ensure that you are familiar with the information in Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings."

To use Fusion Middleware Control to configure a database for the Oracle BI Scheduler:

1. Go to the Business Intelligence Overview page, as described in Section 2.2.2, "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components."
2. Display the Scheduler tab of the Deployment page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
   - **Database** list
   - **Call Interface** list

     The Call Interface is updated automatically according to the Database chosen.
   - **Data Source** option

     For SQL Server databases, you must have a system DSN configured. For information, see Section 25.2.2, "Configuring a System DSN entry for SQL Server Databases."
   - **Username** option

     Enter the user name that was created in the procedure that is described in Section 25.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."
   - **Password** option

     Enter the password of the existing scheduler schema user. For information, see Section 25.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."
   - **Confirm Password** option

     Password details are not checked against the database.
5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

See Chapter 20, "Configuring and Managing Agents" for information about advanced configuring settings for agents.

For information on using Oracle BI Systems Management API methods to change configuration settings, see Chapter 30, "Introducing the Oracle BI Systems Management API."

25.2.2 Configuring a System DSN entry for SQL Server Databases

For SQL Server databases, the Data Source Name (DSN) that is used in the Fusion Middleware Control Scheduler configuration must match an existing ODBC DSN for the SQL Server S_NQ_SCHED database that is used in Section 25.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."

If you do not have a System DSN entry, then create a new one as described in the following procedure.

To configure the SQL Server database DSN entry:

1. From the Windows Start menu, select **Settings**, then **Control Panel**, then **Administrative Tools**, then **Data Sources (ODBC)**.
2. Start the ODBC Data Source Administrator.
3. Select the System DSN tab, and click **Add**.
4. Select the driver SQL Server, and click **Finish**.
5. In the Create a New Data Source to SQL Server wizard, do the following:
   a. Enter a name and description for the data source.
   b. Select the SQL Server from the Server list, and click **Next**.
   c. For server verification of the login ID authenticity, select the appropriate authentication for the S_NQ_SCHED SQL Server database schema that was created in Section 25.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU." Click **Next**.
6. Select the **Change the default database to** field and select the S_NQ_SCHED database from the list. Click **Next**.
7. Update any language or log file settings if appropriate, and click **Finish**.
8. To verify the connection settings, click the **Test Data Source** button, and click **OK**.
9. Click **OK** to exit ODBC Data Source Administrator.
10. Use Fusion Middleware Control to configure the SQL Server database as described in Section 25.2.1, "Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler."

25.3 Configuring Oracle BI Scheduler Instances in a Clustered Environment

In a clustered environment you can have zero, one, or two scheduler instances. Configure zero if you do not require support for scheduled jobs or agents. Configure one if you do not require high availability. Configure two to support high availability.

For information, see Section 6.2.1, "Using Fusion Middleware Control to Configure Primary and Secondary Instances."

25.4 Changing Oracle BI Scheduler Table Names

You can change the names of the tables that the Oracle BI Scheduler uses. This step is required only if the default table names are incompatible with the database setup. Custom table names are configured by adding settings to the Oracle BI Scheduler configuration files for each Scheduler instance. You must make the same changes for each Scheduler instance. Using different table names for the two Scheduler instances results in inconsistent behavior when the active Scheduler changes.

Use various elements in the instanceconfig.xml file to change these settings.

The following procedure describes how to change the Oracle BI Scheduler table names. For this procedure, a new tag, *DB_Column_NamesType,* has been created as an example.

When using database scripts to create the Scheduler tables, the changed table names should be included in the scripts.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually change the settings for the Oracle BI Scheduler table names:

1. Open the Oracle BI Scheduler version of the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"

2. Create a new element named DB_Column_NamesType as shown in the following example from an instanceconfig.xml file:

```
<xs:complexType name="DB_Column_NamesType">
<xs:all>
<xs:element name="TABLE_JOBS" type="xs:string" default="S_NQ_JOB"
minOccurs="0">
  <xs:annotation>
    <xs:documentation xml:lang="en">
     The name of the table used as the jobs table in the back-end DB.
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="TABLE_INSTANCES" type="xs:string" default="S_NQ_INSTANCE"
minOccurs="0">
   <xs:annotation>
     <xs:documentation xml:lang="en">
     The name of the table used as the instances table in the back-end DB.
     </xs:documentation>
   </xs:annotation>
</xs:element>
<xs:element name="TABLE_PARAMS" type="xs:string" default="S_NQ_JOB_PARAM"
minOccurs="0">
<xs:annotation>
  <xs:documentation xml:lang="en">
  The name of the table used as the job parameters table in the back-end DB.
  </xs:documentation>
 </xs:annotation>
</xs:element>
<xs:element name="TABLE_ERRMSGS" type="xs:string" default="S_NQ_ERR_MSG"
minOccurs="0">
  <xs:annotation>
   <xs:documentation xml:lang="en">
   The name of the table used to store information about job instances that do
not complete successfully in the back-end DB.
```

```
   </xs:documentation>
  </xs:annotation>
</xs:element>
```

3. For each of the entries under the tag DB_Column_NamesType, add the parameter and string values that are shown in Table 25–1. The values that are created in the data string become the values that are used for the Oracle BI Scheduler table names.

Table 25–1 Parameters for DB Column Names in instanceconfig.xml

Parameter Name	Type	String Value
TABLE_JOBS	REG_SA	S_NQ_JOB
TABLE_INSTANCES	REG_SA	S_NQ_INSTANCE
TABLE_PARAMS	REG_SA	S_NQ_JOB_PARAM
TABLE_ERRMSGS	REG_SA	S_NQ_ERR_MSG

Note: The data types for each column should remain true to the intent of the schema. For example, if the job ID is defined as an integer type, do not change it to a varchar type. However, increasing the number of characters in a varchar column is an acceptable change.

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

26

Using Oracle BI Scheduler Job Manager

This chapter describes how to manage Oracle BI Scheduler using the Job Manager feature in the Oracle BI Administration Tool. This chapter contains the following topics:

26.1 Opening Job Manager

The Job Manager is a Windows tool that is the interface with the Oracle BI Scheduler. Through Job Manager, you can connect to, start and stop the Oracle BI Scheduler, add and manage jobs, and manage job instances. You create and edit agents using Oracle BI Presentation Services. For information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

The following procedure describes how to open Job Manager from the Oracle BI Administration Tool.

To open the Job Manager:

1. Open the Oracle BI Administration Tool.
2. From the toolbar, click the **Manage** menu, and select **Jobs**.

   Job Manager opens in a new window.
3. In Job Manager, from the toolbar, click the **File** menu, and select **Open Scheduler Connection**.
4. In the Machine Name dialog box, enter the machine name and port number where the Oracle BI Scheduler is located, and click **OK**.

   You can learn the machine name (host) and port number by displaying the Availability tab of the Capacity Management page in Fusion Middleware Control. For information, see Section 6.2.1, "Using Fusion Middleware Control to Configure Primary and Secondary Instances.".

26.2 Adding Oracle BI Scheduler Jobs in Job Manager

Job Manager enables you to add new scripted jobs. Agents can only be created in Oracle BI Presentation Services.

Add an Oracle BI Scheduler job in the Job Manager using the following procedure.

To add an Oracle BI Scheduler job in the Job Manager:

1. In the Job Manager, from the **Jobs** menu, select **Add New Job**.
2. Enter the appropriate information in the dialog box.

   See the following sections for field descriptions:

   - Section 29.3, "General Oracle BI Scheduler Job Properties"
   - Section 29.4, "Job Action Properties Available in Job Manager"
   - Section 29.5, "Job Triggers in Job Manager"

26.3 Modifying Oracle BI Scheduler Jobs in the Job Manager

You can modify an Oracle BI Scheduler job in the Job Manager using the following procedure.

> **Note:** If, while adding or modifying a job in the Job Manager, you enter a script in the Script field, then the Oracle BI Scheduler creates a file with an SCS extension in the following directory:
>
> \orainst\bifoundation\<OracleBISchedulerComponent>\coreapplication_obisch1\scripts\scheduler
>
> Where orainst is the install location for Oracle Business Intelligence.
>
> Oracle BI Scheduler's job scripts are stored in this location (not in the back-end database), so do not remove scripts from here.

To modify an Oracle BI Scheduler job in the Job Manager:

1. In the Job Manager, select the job to modify.
2. From the **Jobs** menu, select **Modify Job**.
3. In the Modify Job dialog box, change the job properties.

   For field descriptions, see Section 29.3, "General Oracle BI Scheduler Job Properties," Section 29.4, "Job Action Properties Available in Job Manager," and Section 29.5, "Job Triggers in Job Manager."

26.4 About Adding Agent Jobs

You cannot add agent jobs using the Job Manager. Agents are most commonly added through Oracle BI Presentation Services. For information, see Section 24.1, "What is Oracle BI Scheduler?"

You can, however, modify an agent job using the Modify Job dialog box. For information, see Section 26.5, "Modifying Agents in Job Manager."

26.5 Modifying Agents in Job Manager

You can modify an individual agent using the Modify Job dialog box in the Job Manager.

To modify an agent in Job Manager:

1. In Job Manager, select the agent to modify.
2. From the **Jobs** menu, select **Modify Job**.
3. In the Modify Job dialog box, change the agent job properties.

   For field descriptions, see Section 29.3, "General Oracle BI Scheduler Job Properties," Section 29.4, "Job Action Properties Available in Job Manager," and Section 29.5, "Job Triggers in Job Manager."

Agent-specific job properties are described in the following table.

> **Note:** A default value in these fields indicates that the value that is specified in Section 20.3.3.3, "Agent Scheduler Configuration Settings" is active.

Agent Property	Description
BI Presentation Server	Specifies the Web server that this agent contacts when it runs. Do not change this setting, because the agent might not exist on a different Web server. This feature was added for debugging purposes only.
Debug Log	Determines whether debugging information is written to a log. This overrides the Scheduler's Debug configuration setting for just this one agent. If set to true, then log files named Agent-<*jobID*>-<*InstanceID*>.log are created in the active Scheduler's log directory when the agent is executed.

26.6 Re-Running a Job Instance

You can re-run any type of job, including an agent job that failed. When you re-run a failed agent job, only the failed items are delivered. For example, if there are 1,000 recipients of an agent, but 200 of those recipients belonged to a group that does not have the appropriate request viewing permissions, then the administrator can correct the permission settings and re-run the job. In this example, only the 200 users in the modified group receive the agent. The agent is not re-delivered to the other 800 users who successfully received the agent.

To re-run a job instance:

1. Open the Oracle BI Administration Tool in Online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.

   The Job Manager window is displayed.
3. Go to the Instance list.
4. Locate the job instance to re-run.
5. Right-click the job instance and select **Re-run Instance(s) Now**.

26.7 Managing Oracle BI Scheduler Job Instances

An *instance* in the Oracle BI Scheduler is a record that stores information regarding a specific execution of an Oracle BI Scheduler job.

You can perform the following tasks using Oracle BI Scheduler job instances:

- Section 26.7.1, "Viewing or Refreshing Oracle BI Scheduler Job Instances"
- Section 26.7.2, "Cancelling Oracle BI Scheduler Job Instances"
- Section 26.7.3, "Purging Oracle BI Scheduler Job Instances"

To work with Oracle BI Scheduler job instances:

1. Click the Instances tab in the lower-left corner of the Job Manager window.
2. When instances are present, use the tree in the left pane to locate instances and view information about them.
   - The Instances menu is described in the table in Section 29.2.4, "Instances Menu in Job Manager."
   - Instances properties are described in the table in Section 29.2.5, "Instance Properties in Job Manager."

26.7.1 Viewing or Refreshing Oracle BI Scheduler Job Instances

You can view Oracle BI Scheduler job instance information using the following procedure.

> **Note:** In some environments, if numerous instances have run and instances have not been purged in some time, then this process can take a few seconds.

To view Oracle BI Scheduler job instance information:

1. Open the Oracle BI Administration Tool in Online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.

   The Job Manager window is displayed.
3. Go to the Instance list.
4. Select a particular job instance and from the **Instances** menu, select **View Instance**.

   A description of the Instance properties shown in the Instance window is given in Section 29.2.4, "Instances Menu in Job Manager."

To refresh Oracle BI Scheduler job instances:

1. In the Instance List, from the **Instances** menu, select **Refresh Instance List**.

26.7.2 Cancelling Oracle BI Scheduler Job Instances

Registered canceled instances are described in Section 27.8.6, "RegisterCancelCommand Method." The cancel event is issued to the Oracle BI Scheduler and the instance is marked as canceled when its registered cancel methods are called.

To cancel an Oracle BI Scheduler job instance:

1. Open the Oracle BI Administration Tool in Online mode.

2. From the **Manage** menu, select **Jobs** from the toolbar.

   The Job Manager window is displayed.

3. Go to the Instance list.
4. Select a particular job instance, and from the **Instances** menu, select **Cancel Instance(s)**.

26.7.3 Purging Oracle BI Scheduler Job Instances

Purging a job instance involves removing it from the back-end database using one of the following methods:

- Section 26.7.3.1, "Using the Job Manager Instances List"
- Section 26.7.3.2, "Using the Job Manager Purge Instances Window"

26.7.3.1 Using the Job Manager Instances List

The following procedure purges Oracle BI Scheduler job instances through the Instances List.

To purge Oracle BI Scheduler job instances through the Instances List:

1. Open the Oracle BI Administration Tool in online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.

   The Job Manager window is displayed.

3. Go to the Instance list.
4. Select the instances from the Instance List and click **Delete**.

26.7.3.2 Using the Job Manager Purge Instances Window

The following procedure purges job instances through the Purge Instances window.

To purge Oracle BI Scheduler job instances through the Purge Instances window:

1. Open the Oracle BI Administration Tool in online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.

   The Job Manager window is displayed.

3. Go to the Instance list.
4. Click the Purge Instance(s) icon on the toolbar or from the **Instances** menu, select **Purge Instances** to open the Purge Instances window.

   You can purge instances by JobID, by UserID, or by End Time.

   If you select the End Time method, then all jobs with an End Time less than or equal to the given time are purged.

5. Select the purge method to use.
6. Click **OK** when you have finished to return to the Job Manager window.

27 Programming BI Scheduler VBScript and JScript Jobs

You can use the Oracle BI Scheduler to schedule general purpose scripts that extend the functionality of Oracle Business Intelligence.

Scripts can either be standalone Script Jobs (in Job Manager), or Script Actions tagged onto the end of agents. (For more information on agents, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*). Both types of script use the same script facilities, with a few exceptions. For example Script Actions can access the result set that is being delivered by the agent, while standalone Script Jobs have no access to result sets.

Oracle BI Scheduler includes a Script object that encapsulates a running script. The Script object represents a script and exposes the properties and methods of a script. You can access its methods and properties directly because its name is implied. For example, to access the JobID property, you can specify JobID, not Script.JobID.

> **Note:** Scripting for agents and scripts defined by Oracle BI Scheduler Job Manager are supported only under Windows platforms and are not supported under UNIX.

This chapter describes how to configure custom script properties, includes scripting examples, and provides detailed information about script job properties. It contains the following topics:

- Section 27.1, "Configuring Custom Script Properties for Oracle BI Scheduler"
- Section 27.2, "Creating a Custom Script Example - Cache Clearance"
- Section 27.3, "Configuring Custom Script Properties for Oracle BI Delivers"
- Section 27.4, "Creating a Custom Script Example - Copy Results to the File System"
- Section 27.5, "Oracle BI Scheduler Read-Only Script Object Properties"
- Section 27.6, "Oracle BI Scheduler Read/Write Script Object Properties"
- Section 27.7, "Oracle BI Scheduler Script-Defined Constants"
- Section 27.8, "Oracle BI Scheduler Script Object Methods and Events"
- Section 27.9, "Troubleshooting JScript and VBScript Job Failures"

27.1 Configuring Custom Script Properties for Oracle BI Scheduler

Use the following procedure to modify the properties of an existing Oracle BI Scheduler script.

To add this job as a standalone job in Job Manager, see Section 26.2, "Adding Oracle BI Scheduler Jobs in Job Manager."

> **Note:** The script has to exist on the Oracle BI Scheduler server computer before you can configure the properties.

To configure custom Oracle BI Scheduler script properties:

1. Set the custom properties according to Section 29.4, "Job Action Properties Available in Job Manager."

   For example, for the purgeSASCache.js script, use the values that are shown in the following table. To view an example of the SASCache.js script, see Section 27.2, "Creating a Custom Script Example - Cache Clearance."

Field	Value or Setting
Script Type	JScript
Script / Script File check boxes	Script File
Script	purgeSASCache.js
Parameter(0): User	Administrator
Parameter(1): Password	*your_password*

2. Click **OK**.

27.2 Creating a Custom Script Example - Cache Clearance

You can use the purgeSASCache.js script to periodically purge all of the cache from the Oracle BI Server. The file must be saved in the following directory:

ORACLE_INSTANCE\bifoundation\OracleBISchedulerComponent\coreapplication_obisch*n*\scripts\common

```
/////////////////////////////////////////////////////////
//purgeSASCache.js
//
//Purges the cache on SAS.
//Parameter(0) - The user name to pass in to NQCMD.
//Parameter(1) - The password for the aforementioned user.
/////////////////////////////////////////////////////////
//The full path to nqcmd.exe
var nqCmd = "[$INSTALLDIR]\\server\\Bin\\nqcmd.exe";
//The data source name
var dsn = "BI Web";
//The user to execute the queries
var user = Parameter(0);
//The password of the aforementioned user
var pswd = Parameter(1);
//The ODBC procedure call for purging the cache
var sqlStatement = "{call SAPurgeAllCache()};";
//////////////////////////////////////////////////////////
```

```
//Returns a string from the file name
/////////////////////////////////////////////////////////
function GetOutput(fso, fileName)
{
    var outStream = fso.OpenTextFile(fileName, 1);
    var output = outStream.ReadAll();
    outStream.Close();
    return output;
}
/////////////////////////////////////////////////////////
// Get WshShell object and run nqCmd. Capture the output
// so that we can handle erroneous conditions.
var wshShell = new ActiveXObject("WScript.Shell");
// Create a temp file to input the SQL statement.
var fso = new ActiveXObject("Scripting.FileSystemObject");
var tempFolder = fso.GetSpecialFolder(2);
var tempInFileName = fso.GetTempName();
var tempOutFileName = fso.GetTempName();
tempInFileName = tempFolder + "\\" + tempInFileName;
tempOutFileName = tempFolder + "\\" + tempOutFileName;
var tempInFile = fso.CreateTextFile(tempInFileName, true);
tempInFile.WriteLine(sqlStatement);
tempInFile.Close();
try
{
    // execute
    var dosCmd = nqCmd + " -d \"" + dsn + "\" -u \"" + user
        + "\" -p \"" + pswd + "\" -s \"" + tempInFileName + "\"" +
        " -o \"" + tempOutFileName + "\"";
    wshShell.Run(dosCmd, 0, true);
    var output = GetOutput(fso, tempOutFileName);
    // Remove the temp files
    fso.DeleteFile(tempInFileName);
    if (fso.FileExists(tempOutFileName)) {
        fso.DeleteFile(tempOutFileName);
}
    // Check the output for any errors
    if (output.indexOf("Processed: 1 queries") == -1) {
        ExitCode = -1;
        throw Error(-1, output);
    }
    else if (output.indexOf("Encountered") != -1) {
        ExitCode = -2;
        throw Error(-2, output);
    }
        ExitCode = 0;
} catch (e) {
    if (fso.FileExists(tempInFileName)) {
        fso.DeleteFile(tempInFileName);
    }
    if (fso.FileExists(tempOutFileName)) {
        fso.DeleteFile(tempOutFileName);
    }
    throw e;
}
```

27.3 Configuring Custom Script Properties for Oracle BI Delivers

You set script properties on the Actions tab of an agent in Oracle BI Delivers. See *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for details. Refer also to Section 26.5, "Modifying Agents in Job Manager."

> **Note:** The script has to exist on the Oracle BI Scheduler server computer before you can create the agent. Create the custom script, and then create the agent to call the script. See Section 27.4, "Creating a Custom Script Example - Copy Results to the File System."

To configure custom script properties for agents:

1. On the Home page in Oracle BI EE, click the New menu and select the **Agent** option.
2. Display the Conditions tab and select the **Use a condition** box.
3. Click **Create** and **Browse** to select an analysis.
4. Click **OK**.
5. Click **OK**.
6. Display the Actions tab.
7. Click the **Add New Action** icon and select the **Invoke Server Script** menu option.

   The New Action - Invoke Server Script dialog is displayed.
8. Click the **Add Document Parameter** icon.
9. Select the first row of parameters and click the **Delete** button.
10. Enter properties for the parameter displayed.

    For example, for the script that is shown in Section 27.4, "Creating a Custom Script Example - Copy Results to the File System," you use the values that are described in the following table.

Field	Value or Setting
Language	JavaScript
Script Path	createResultfile.js
Name	Enter a name for the parameter.
Prompt	Enter a prompt. For example "Result Set:"
Value	Select Value from the list. This maps a fixed value, for example, PDF. Select Condition Analysis
Optional	Clear the check box.

11. Use the plus icon to display a new row for a second parameter.
12. Enter Result Set Extension in the **Prompt** field.
13. Enter .pdf into the **Value** field.
14. Click **OK**.
15. Save the agent.

This script runs after the Conditional Request of the agent.

27.4 Creating a Custom Script Example - Copy Results to the File System

This example configures a script for the Oracle BI Scheduler that copies the results of an agent to another directory. The script copies the temporary file that contains the results of the Conditional Request to the agent log directory. The JobID, InstanceID, and UserID are used in the file name to guarantee that the result sets do not overwrite each other with each execution of the agent, for each user, or for other agents that share this script.

To add this job in an agent, see Section 27.3, "Configuring Custom Script Properties for Oracle BI Delivers."

The example script uses the following values:

- The agent log directory on the Oracle BI Scheduler computer is $OracleBI\instances\logs\Agents (where $OracleBI is the location in which the Oracle Business Intelligence software is installed).
- The agent is run as Administrator.
- The Custom Script properties are set according to the table in Section 27.3, "Configuring Custom Script Properties for Oracle BI Delivers."
- If the job ID is 101 and instance ID is 1208, then you see a file called 101-1208-Administrator-AgentScript1.pdf.

The output of this example, after the agent is run, is a file on the Oracle BI Scheduler computer called D:\OracleBI\Log\Agents\101-1208-Administrator-AgentScript1.PDF.

This file contains the results of the Conditional Request in PDF format.

For all script jobs from chained agents, the full path name to the temporary file is specified in Parameter(0).

```
/////////////////////////////////////////////////////////////
//
// createResultFile.js
//
// Copies the results in the temporary file to a new file name
//
// Parameter(0) = Agent Result File Path
// Parameter(1) = Last Part of Output File Name (no path)
//
/////////////////////////////////////////////////////////////
var FSO = new ActiveXObject("Scripting.FileSystemObject");
var fileName = GetConfigurationValue("Log Dir", "Agents") +
    "\\" + JobID + "-" + InstanceID + "-" + UserID + "-" +
    Parameter(1);
var fooFile = FSO.CopyFile(Parameter(0), fileName, true);
```

27.5 Oracle BI Scheduler Read-Only Script Object Properties

The Oracle BI Scheduler supports the read-only script object properties that are described in Table 27–1.

Table 27–1 Oracle BI Scheduler Read-Only Script Object Properties

Object Property	Description	Return Value	Syntax
JobID	Returns the job identification number that is associated with this instance.	long	NA
InstanceID	Returns the instance identification number that is associated with this instance.	long	NA
ParameterCount	Returns the number of job parameters that is associated with the job script.	long	NA
Parameter (index)	Returns a specific parameter that is associated with the script. Parameter (index) returns an error if the given index is less than zero or greater than ParameterCountminus 1.	string	Parameter(index) Index is the zero-based index of the parameter.
Script	Returns the Script object that represents the current script. This object implements the COM IDispatch interface and can be passed as arguments to methods of other objects that exist on the system. Implementing the COM IDispatch is particularly useful when handling cancel events to a running instance. See Section 27.8.6, "RegisterCancelCommand Method".	script object	NA
UserID	Returns the user identification number that is associated with the instance.	string	NA

27.6 Oracle BI Scheduler Read/Write Script Object Properties

The Oracle BI Scheduler supports the read/write script object properties that are shown in Table 27–2.

Table 27–2 Oracle BI Scheduler Read/Write Script Object Properties

Object Property	Description	Return Value
Message	Sets or returns the Message property of the running instance. The Message property can convey meaningful error information. Setting this value changes the Message field of a Job Instance without stopping execution of the current Job Script. If the JScript throw() method is called and this property has been set, then the value is appended to the message description in the JScript or VBScript Error object. COM objects that implement the IDispatch interface can be accessed from within Job Scripts. If any method fails and properly provides error information through the SetErrorInfo() method, then that information is contained in the Message field of the Job Instance. If the Message property is set before the COM object error is generated, then that string value is appended to the COM object error information.	string
Severity	Sets the instance status. You can set it to any of the Severity Constants, as described in Section 27.7.1, "Severity Constants.". By default, it is set to nqSeverityInformation.	string
ExitCode	Sets or returns the Exit Code property that is associated with the instance. The default is 0 (zero). See the description of ExitCode instance properties in Section 29.2.5, "Instance Properties in Job Manager."	long

27.7 Oracle BI Scheduler Script-Defined Constants

The Oracle BI Scheduler supports the following script-defined constants. These constants are used by the methods to schedule new jobs.

- Section 27.7.1, "Severity Constants"
- Section 27.7.2, "DayEnum Constants"
- Section 27.7.3, "DayOfWeekEnum Constants"
- Section 27.7.4, "JobFlagsEnum Constants"
- Section 27.7.5, "MonthEnum Constants"
- Section 27.7.6, "OccurrenceEnum Constants"

27.7.1 Severity Constants

This topic is part of Section 27.7, "Oracle BI Scheduler Script-Defined Constants."

Severity constants are used in the severity property of a Message (error message) returned by a script to determine the status of a job instance. Table 27–3 describes Severity values.

Table 27–3 Severity Constant Values

Value	Description
nqSeverityInformation	Set the Severity property to *Information* if the Message contains only information for the job instance; that is, no error condition is reflected. The status of the instance is set to Completed. This is the default if Severity is not set.
nqSeverityWarning	Set the Severity property to *Warning* if the Message contains text that describes a non-critical failure. The instance status is set to Warning.
nqSeverityError	Set the Severity property to *Error* if the Message contains text that indicates a critical failure. The instance status is set to Failed.

27.7.2 DayEnum Constants

This topic is part of Section 27.7, "Oracle BI Scheduler Script-Defined Constants."

The DayEnum values are used with the scheduling functions to identify days in a month, from Day 1 to Day 31. Table 27–4 describes DayEnum values.

Table 27–4 DayEnum Constant Values

Value	Description
nqDay1	Day 1
nqDay2	Day 2
nqDay3	Day 3
...	...
nqDay31	Day 31

27.7.3 DayOfWeekEnum Constants

This topic is part of Section 27.7, "Oracle BI Scheduler Script-Defined Constants."

The DayOfWeekEnum values are used with the scheduling functions to identify days in a week. Table 27–5 describes DayOfWeekEnum values.

Table 27–5 DayOfWeekEnum Constant Values

Value	Description
nqSunday	Sunday
nqMonday	Monday
nqTuesday	Tuesday
nqWednesday	Wednesday
nqThursday	Thursday
nqFriday	Friday
nqSaturday	Saturday

27.7.4 JobFlagsEnum Constants

This topic is part of Section 27.7, "Oracle BI Scheduler Script-Defined Constants."

The JobFlagsEnum values are used with the scheduling methods of the Script object to control how a job behaves. Table 27–6 describes JobFlagsEnum values.

Table 27–6 JobFlagsEnum Constant Values

Value	Description
nqJobNoFlags	This flag indicates that the job has no special behavior.
nqJobDeleteWhenDone	This flag indicates that the job is deleted when there are no more scheduled run times.
nqJobDisabled	This flag indicates that the job is disabled. This is useful for preventing a job from running at the scheduled time or times.
nqJobHasEndDate	This flag indicates that the job has a valid end date.
nqJobExecuteWhenMissed	If for some reason the Oracle BI Scheduler is down when the job is supposed to start, then this flag indicates that the job should run when the Oracle BI Scheduler starts again.
nqJobDeleteScriptWhenDone	When a job is removed and this flag is set, the script that is associated with the job is deleted. This is useful only with the nqJobScriptContainsPath flag.
nqJobScriptContainsPath	This flag indicates that the script that is associated with the job contains a path to a file that contains the actual script code.
nqJobStartNow	When this flag is set, the begin date and start time are ignored. Instead, these fields are set to the current time of the Oracle BI Scheduler.

27.7.5 MonthEnum Constants

This topic is part of Section 27.7, "Oracle BI Scheduler Script-Defined Constants."

The MonthEnum values are used with the scheduling functions to identify months. Table 27–7 describes MonthEnum values.

Table 27–7 MonthEnum Constant Values

Value	Description
nqJanuary	January
nqFebruary	February
nqMarch	March
nqApril	April
nqMay	May
nqJune	June
nqJuly	July
nqAugust	August
nqSeptember	September
nqOctober	October
nqNovember	November
nqDecember	December

27.7.6 OccurrenceEnum Constants

This topic is part of Section 27.7, "Oracle BI Scheduler Script-Defined Constants."

The OccurrenceEnum values are used with the scheduling functions to identify the occurrence of a given day. Table 27–8 describes OccurrenceEnum values.

Table 27–8 OccurrenceEnum Constant Values

Value	Description
nqFirst	First occurrence
nqSecond	Second occurrence
nqThird	Third occurrence
nqFourth	Fourth occurrence
nqLast	Last occurrence

27.8 Oracle BI Scheduler Script Object Methods and Events

You use script object methods and events for the Oracle BI Scheduler when writing programs, as described in Chapter 27, "Programming BI Scheduler VBScript and JScript Jobs." The following sections describe methods and events:

- Section 27.8.1, "CreateArray Method"
- Section 27.8.2, "DeregisterCancelCommand Method"
- Section 27.8.3, "GetConfigurationValue Method"
- Section 27.8.4, "GetTempFileName Method"
- Section 27.8.5, "LaunchProcess Method"
- Section 27.8.6, "RegisterCancelCommand Method"
- Section 27.8.7, "ScheduleJobDaily Method"
- Section 27.8.8, "ScheduleJobMonthlyDate Method"

- Section 27.8.9, "ScheduleJobMonthlyDOW Method"
- Section 27.8.10, "ScheduleJobNow Method"
- Section 27.8.11, "ScheduleJobOnce Method"
- Section 27.8.12, "ScheduleJobWeekly Method"
- Section 27.8.13, "OnError Event"

27.8.1 CreateArray Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Creates an Array object.

Usage: This method is provided only for JScript because local JScript Array objects cannot be passed directly to the Script methods. This method is called to create an array object and to pass the array object to Script methods that accept an array as an argument.

Syntax 1: Set array = CreateArray ()

Syntax 2: Set array = CreateArray (size)

Syntax 3: Set array = CreateArray (element 0, element 1, ..., element *n*)

The different syntax versions create arrays as follows:

- Syntax 1 creates an array of size 0 (zero).
- Syntax 2 creates an array with the specified size.
- Syntax 3 creates an array filled with the specified elements.

Example 27–1 Example

```
var i;
var array1= CreateArray(2);
for (i = 0; i < array1.Size; i++)
{
   array1(i) = i;
}

   array1.Resize(4);
for (i = 2; i < array1.Size; i++)
{
   array1(i) = i;
}

var array2 = CreateArray(0, 1, 2,3);
for (i = 0; i < array2.Size; i++)
{
   if (array1(i) != array2(i))
   break;
}
```

Arguments: See Table 27–9 for CreateArray method arguments.

Return Value: Returns an Array object.

Table 27–9 CreateArray Method Arguments

Argument	Description
size	A long value that specifies the initial size of the array.
element0 … elementn	The values to place in the array. This creates an array with the lower and upper bounds of 0 (zero) and n, respectively.

27.8.2 DeregisterCancelCommand Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Deregisters a previously registered cancel method.

Usage: Call this method to deregister the most recently registered cancel method after a long operation has completed successfully. You need not call this method if the script was canceled.

Syntax: `DeregisterCancelCommand`

27.8.3 GetConfigurationValue Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Returns the value in the configuration relative to the root registry entry of the Oracle BI Scheduler.

Usage: Returns the string value for a registry setting relative to the Oracle BI Scheduler. The `configKey` and `subkeyPath` strings must be identical to those in the registry.

Syntax: value = `GetConfigurationValue(`configKey `[,` subkeyPath`])`

Arguments: See Table 27–10 for GetConfigurationValue method arguments.

Return Value: Returns a string value.

Table 27–10 GetConfigurationValue Method Arguments

Argument	Description
configKey	A string that specifies the registry key name to return.
subkeyPath	(Optional) A string value that specifies the registry path below the Oracle BI Scheduler's root path.

27.8.4 GetTempFileName Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Returns a temporary file name.

Usage: GetTempFileName() does not create a file. It only provides a temporary file name for use in creating a file. Files that are created in job scripts are not deleted automatically when the script terminates.

Syntax: `tfname = GetTempFileName()`

Return Value: Returns a string value.

27.8.5 LaunchProcess Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Executes a command line in a new process.

Usage: Call this method to execute a command line in a new process. If the wait argument is set to True, then this method returns the exit code that is returned by the process.

Syntax: `exitcode = LaunchProcess ( commandLine [, wait, terminateOnCancel] )`

Arguments: See Table 27–11 for LaunchProcess method arguments.

Return Value: Returns a long value.

Table 27–11 LaunchProcess Method Arguments

Argument	Description
commandLine	A string that specifies the command line to execute.
wait	(Optional) A Boolean value that specifies whether the method should wait for the process to terminate. The default is True.
terminateOnCancel	(Optional) A Boolean value that specifies whether the method should terminate the process when the script is canceled. The default is True.

27.8.6 RegisterCancelCommand Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Registers a method to be called when the script is canceled.

Usage: Occasionally, an object's method takes a long time to complete. If the job is canceled before the call returns, then the script engine still must wait until the call returns. This could potentially take hours and limit resources. This method solves the problem by registering a method that is asynchronously called by the script engine if the script gets canceled.

Cancel methods should be registered before calling the method that executes a long operation. When the method returns, the cancel method should be deregistered by calling DeregisterCancelCommand().

A good practice is to hide implementation details of a COM object from the caller, having the COM object itself handle all registration and deregistration of cancel commands. Pass an instance of the Script object to the COM object, then call the RegisterCancelCommand() and DeregisterCancelCommand() methods because the Script object implements the IDispatch interface.

Syntax: `RegisterCancelCommand source, methodName [, arguments]...`

Arguments: See Table 27–12 for RegisterCancelCommand method arguments.

Table 27–12 RegisterCancelCommand Method Arguments

Argument	Description
source	An object whose method is being registered.
methodName	A string that specifies the method name.

Table 27–12 (Cont.) RegisterCancelCommand Method Arguments

Argument	Description
arguments	Optional arguments to be passed into the method.

27.8.7 ScheduleJobDaily Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Schedules a new job with a Daily trigger.

Syntax: `ScheduleJobDaily name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, daysInterval [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]`

Arguments: See Table 27–13 for ScheduleJobDaily method arguments.

Table 27–13 ScheduleJobDaily Method Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
daysInterval	An integer value that specifies the number of days between job invocations.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see Section 27.7.4, "JobFlagsEnum Constants". The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

27.8.8 ScheduleJobMonthlyDate Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Schedules a new job with a Monthly by Date trigger.

Syntax: `ScheduleJobMonthlyDate name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, whichDays, whichMonths [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]`

Arguments: See Table 27–14 for ScheduleJobMonthlyDate method arguments.

Table 27–14 ScheduleJobMonthlyDate Method Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
whichDays	An long value that specifies the days of the month on which the job runs. For valid settings, see Section 27.7.2, "DayEnum Constants."
whichMonths	An integer value that specifies the months in which the job runs. For valid settings, see Section 27.7.5, "MonthEnum Constants."
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see Section 27.7.4, "JobFlagsEnum Constants." The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

27.8.9 ScheduleJobMonthlyDOW Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Schedules a new job with a monthly by day of the week (DOW) trigger.

Syntax: `ScheduleJobMonthlyDOW name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, whichOccurrences, whichDays, whichMonths [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]`

Arguments: See Table 27–15 for ScheduleJobMonthlyDOW method arguments.

Table 27–15 ScheduleJobMonthlyDOW Method Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
whichOccurrences	An integer value that specifies the occurrences of days of the week on which the job runs. For valid settings, see Section 27.7.2, "DayEnum Constants".
whichDays	An integer value that specifies the days of the week on which the job runs. For valid settings, see Section 27.7.3, "DayOfWeekEnum Constants".
whichMonths	An integer value that specifies the months in which the job runs. For valid settings, see Section 27.7.5, "MonthEnum Constants".
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see Section 27.7.4, "JobFlagsEnum Constants". The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

27.8.10 ScheduleJobNow Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Schedules a new job with a Run Now trigger.

Syntax: `ScheduleJobNow name, description, scriptType, script [, parameters, flags, maxRunTimeMS]`

Arguments: See Table 27–16 for ScheduleJobNow method arguments.

Table 27–16 ScheduleJobNow Method Arguments

Argument	Description
name	A string that specifies the name of the job.

Table 27–16 (Cont.) ScheduleJobNow Method Arguments

Argument	Description
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see Section 27.7.4, "JobFlagsEnum Constants". The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

27.8.11 ScheduleJobOnce Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Schedules a new job with a Run Once trigger.

Syntax: `ScheduleJobOnce name, description, scriptType, script, startDate, startTime [,` parameters, flags, `maxRunTimeMS]`

Arguments: See Table 27–17 for ScheduleJobOnce method arguments.

Table 27–17 ScheduleJobOnce Method Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see Section 27.7.4, "JobFlagsEnum Constants". The default is nqJobNoFlag.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

27.8.12 ScheduleJobWeekly Method

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Schedules a new job with a Weekly trigger.

Syntax: `ScheduleJobWeekly name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, weeksInterval, whichDays [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]`

Argument: See Table 27–18 for ScheduleJobWeekly method arguments.

Table 27–18 ScheduleJobWeekly Method Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
weeksInterval	An integer value that specifies the number of weeks between job invocations.
whichDays	An integer value that specifies the days of the week on which the job runs. See Section 27.7.3, "DayOfWeekEnum Constants" for valid settings.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see Section 27.7.4, "JobFlagsEnum Constants". The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

27.8.13 OnError Event

This topic is part of Section 27.8, "Oracle BI Scheduler Script Object Methods and Events."

Occurs when the script engine encounters a run-time error while executing the script. This is intended for cleanup purposes, but the creative use of try/catch blocks in JScript and appropriate Error Handling in VBScript are often superior alternatives to using this event.

Usage: The script engine calls this procedure when it encounters a run-time error while executing the script. Define this procedure in your script to perform cleanup activities before the script terminates, such as deleting temporary files and releasing resources.

Syntax: `OnError`

Example 27–2 Using VBScript:

```
Public Sub OnError()
   LogFile.WriteLine "Encountered a runtime error in the script."
LogFile.Close
End Sub
```

Example 27–3 Using JScript:

```
function OnError()
{
   LogFile.WriteLine("Encountered a runtime error in the
   script.");
LogFile.Close();
```

27.9 Troubleshooting JScript and VBScript Job Failures

If a JScript or VBScript job fails with the error "nQSError: 66001] Failed to create the ActiveX scripting engine.", then the required script engine (VBScript or JScript) is not available or is broken on this computer.

You can reregister the script DLL files in Windows by running the command "`regsvr32 vbscript.dll`" for a VBScript failure or "`regsvr32 jscript.dll`" for a JScript failure. If these return the message "failed module could not be found", then you must repair the Windows installation to re-instate the missing DLL files. You can achieve this by rolling back to a restore point, or by carrying out a repair using the original operating system installation discs.

28 Programming BI Scheduler Java Jobs

You can use the Oracle BI Scheduler to schedule Java programs, called Java jobs, that extend the functionality of Oracle Business Intelligence. Java jobs can be either standalone Scheduler Jobs in Job Manager (Windows), or existing Java Actions in Oracle BI Delivers that are added to the end of agents and upgraded from Release 10*g* (Windows or UNIX).

Note that existing Java actions that are upgraded from Release 10*g* can be executed, but not created in Release 11*g*. Instead, you should use the EJB-based Java actions that are available in this release. For more information on actions, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

This chapter describes programming Java jobs for the Oracle BI Scheduler and contains the following topics:

28.1 Using Oracle BI Scheduler Java Jobs

Java jobs are Java programs that are executed by the JavaHost process on behalf of the Oracle BI Scheduler. Java jobs are different than Java EJB actions. A Java job is deployed in a JAR file, with the entry point defined by one class. That class must implement the SchedulerJavaExtension interface. The job's context is provided by the input SchedulerJobInfo parameter.

This section contains the following information:

28.1.1 Adding and Configuring Custom Java Jobs in Job Manager

You add and configure custom Java jobs in the Modify Job and Add Jobs dialog boxes in Job Manager. Refer to Section 26.2, "Adding Oracle BI Scheduler Jobs in Job Manager", and Section 26.3, "Modifying Oracle BI Scheduler Jobs in the Job Manager".

> **Note:** The Java program must exist on the Oracle BI Scheduler server computer before you can create the job in Job Manager. Create the Java program, and then create the job to call the Java program.

To configure and add custom Java jobs in Job Manager:

1. Set the custom properties according to the descriptions in Section 29.4, "Job Action Properties Available in Job Manager".
2. In the Add New Job window, enter the properties.

   For example, for the Java program filecopy.jar, use the values that are shown in the following table. To view an example of the filecopy.jar program, see Section 28.1.2, "Example: Creating a Java Program for Agents".

Field	Value or Setting
Script Type	Java
Class Name	sched.sched The Java class that you created in Section 28.1.2, "Example: Creating a Java Program for Agents."
Class Path	filecopy.jar The JAR file that contains the Java class.
Parameters	c:\tmp\report.pdf **Note:** The owner of the JavaHost process must have write permissions on this directory point.

3. Click **OK**.

   The Java program is run after the Conditional Request of the agent is run.

   You cannot add a new Java job action to an agent in Oracle BI Delivers. You can use only existing ones that have been upgraded to this release. However, you can add a new Java job to a job in Job Manager. For information, see Section 26.2, "Adding Oracle BI Scheduler Jobs in Job Manager."

28.1.2 Example: Creating a Java Program for Agents

This example creates a Java program that copies the results of an agent to another directory. The example creates a Java class that contains filecopy logic.

To create a Java program to be used with agents:

1. Create a Java program using a Java editor.
   a. Create a new Java class called 'sched'.
   b. Paste the following code into the Java editor:

```
package sched;
    import java.io.*;
    import java.lang.Thread;
    import
com.siebel.analytics.scheduler.javahostrpccalls.SchedulerJavaExtension;
    import
com.siebel.analytics.scheduler.javahostrpccalls.SchedulerJobException;
    import
com.siebel.analytics.scheduler.javahostrpccalls.SchedulerJobInfo;

    public class sched implements SchedulerJavaExtension{
    public void run(SchedulerJobInfo jobInfo) throws SchedulerJobException
    {
      System.out.println("JobID is:" + jobInfo.jobID());
      System.out.println("Instance ID is:" + jobInfo.instanceID());
      System.out.println("JobInfo to string is:" + jobInfo.toString());
      try
      {
        // File outputFile = new File("D:\\JavaJob.txt");
        File attachFile = jobInfo.getResultSetFile();

        InputStream in = new FileInputStream(attachFile.getAbsolutePath());
      OutputStream out = new FileOutputStream(jobInfo.parameter(0));
        byte[] buf = new byte[1024];
        int len;
        while ((len = in.read(buf)) > 0)
        {
          out.write(buf, 0, len);
        }
        in.close();
        out.close();

      }
    catch(Exception ex)
    {
      throw new SchedulerJobException(1, 1, ex.getMessage());
    }
    }
    public void cancel()
    {
    }
    }
```

c. Add the schedulerrpccalls.jar file from the \MW_HOME\ORACLE_HOME\bifoundation\javahost\lib\scheduler directory into your classpath.

d. Compile the Java Class without errors.

e. Jar the compiled output to a file. For example, filecopy.jar.

f. Note the location of the file and ensure that there are no errors.

28.1.3 Example: Configuring a Java Program

This example configures the Java program that you created in Section 28.1.2, "Example: Creating a Java Program for Agents" to enable the Java job to work with agents.

To configure a Java program to be used with agents:

1. Copy the filecopy.jar file that you created in Section 28.1.2, "Example: Creating a Java Program for Agents" to the following directory:

   ORACLE_HOME\bifoundation\javahost\lib

2. Make the following changes to the JavaHost configuration file, which is called config.xml:

   ```
   <Scheduler>
     <Enabled>True</Enabled>
   <DefaultUserJarFilePath>D:\<ORACLE_
   HOME>\bifoundation\javahost\lib</DefaultUserJarFilePath>
   </Scheduler>
   ```

   See Section E.2, "Using the JavaHost Service for Oracle BI Presentation Services" for information on working with the JavaHost configuration file.

   If the JavaHost file is not configured correctly, then the agent log file can stop getting written to, although the agent and the Scheduler are still running. In this situation, you stop the Scheduler using the Windows Task Manager.

3. Restart the JavaHost service.

28.2 Oracle BI Scheduler Java Jobs

The Oracle BI Scheduler integrates with the JavaHost Service to support a custom Java program. The Oracle BI Scheduler provides two Java interfaces (SchedulerJavaExtension and SchedulerJobInfo) and one Java class (SchedulerJobException). You provide a class that implements the SchedulerJavaExtension interface.

> **Note:** For more information about the JavaHost service, see Section E.2, "Using the JavaHost Service for Oracle BI Presentation Services."

28.3 Adding Java Jobs for Oracle BI Scheduler

Use the following procedure to add a Java job for the Oracle BI Scheduler.

> **Note:** The compiled Java class file has to exist on the JavaHost computer before you can configure the properties.

To add a Java Job for Oracle BI Scheduler:

1. Access the Job Manager and from the Jobs menu, select **Add New Job**.

   The Add New job window appears.

2. In the Script Type field, select Java.

3. Specify the custom properties. For information about setting these values, see Section 29.4, "Job Action Properties Available in Job Manager".

   Example values and settings for a Java job with the class name "sample.Test", file path "Sample", and no additional paths and parameters are included below.

Field	Value or Setting
Script Type	Java
Class Name	sample.Test
Class File (Jar File)	Sample

4. Click **OK**.

28.4 Oracle BI Scheduler Custom Java Program Package

The public interfaces and class for Oracle BI Scheduler Custom Java Program are packaged as *com.siebel.analytics.scheduler.javahostrpccalls*. There are two interfaces and one class, which are described in following topics:

- Section 28.5, "SchedulerJavaExtension Interface"
- Section 28.6, "SchedulerJobInfo Interface"
- Section 28.7, "SchedulerJobException Class"

28.5 SchedulerJavaExtension Interface

Your custom code must implement the following interface:

```
package com.siebel.analytics.scheduler.javahostrpccalls;
public interface SchedulerJavaExtension {
public void run(SchedulerJobInfo jobInfo) throws SchedulerJobException;
public void cancel();
}
```

This interface has two methods: run and cancel. The following table describes the methods:

Method	Description
run	This method is invoked by the JavaHost. It provides one SchedulerJobInfo object (described below), which contains instance-related properties such as user ID, Job ID, and Instance ID and parameters. The method is declared to throw SchedulerJobException, which is also described below.
cancel	This method is invoked if the Job instance is still running while Scheduler wants to cancel it. The cancel method is called concurrently by a different thread. Your implementation must therefore protect any data that is shared by the run and cancel methods by synchronization blocks. A typical implementation would be to set a 'cancelCalled' boolean in the cancel method implementation, and check this in any long running loops in the run implementation.

28.6 SchedulerJobInfo Interface

The SchedulerJobInfo interface provides information about the currently running job instance to the custom code:

```
package com.siebel.analytics.scheduler.javahostrpccalls;
import java.io.*;
public interface SchedulerJobInfo {
public final int kJavaJobInformation = 0;
public final int kJavaJobWarning = 1;
public final int kJavaJobError = 2;
int jobID();
int instanceID();
int parameterCount();
String parameter(int index);
boolean hasResultSet();
File getResultSetFile();
String userID();
int getExitCode();
void setExitCode(int exitCode);
int getStatus();
void setStatus(int status);
String getMessage();
void setMessage(String message);
void appendMessage(String message);
}
```

Three public final integers, *kJavaJobInformation*, *kJavaJobWarning*, and *kJavaJobError* are the suggested values that are used to set the status depending upon the circumstances. The following table describes the circumstances:

Members	Description
public final int kJavaJobInformation = 0	Contains an informational message.
public final int kJavaJobWarning = 1	Contains a warning message.
public final int kJavaJobError = 2	Contains an error message.

The following table describes all the methods that are declared in the interface:

Method	Description
int jobID()	Returns the job ID that is associated with the agent.
int instanceID()	Returns the instance ID that is associated with the agent.
int parameterCount()	Returns how many parameters are associated with the agent.
String parameter(int index)	Returns the indexed parameter for the agent.(1).
boolean hasResultSet()	Specifies if there is a result set for this agent.
File getResultSetFile()	Returns a file of result set for this agent (2).
String userID()	Returns the ID of the user who is running the agent.
int getExitCode()	Returns the exit code for the agent.
void setExitCode(int exitCode)	User can set the exit code for the agent.
int getStatus()	Returns the status code for the agent.
void setStatus(int status)	User can set the status code for the agent.
String getMessage()	Returns the message that is associated with the agent.
void setMessage(String message)	User can set the message that is associated with the agent. It replaces the existing message.
void appendMessage(String message)	User can append an additional message to the agent.

28.7 SchedulerJobException Class

If your custom code cannot complete successfully, then throw an instance of this exception class.

```
package com.siebel.analytics.scheduler.javahostrpccalls;
public final class SchedulerJobException extends Exception {
public SchedulerJobException(int exitCode, int status, String message) {
m_exitCode = exitCode;
m_status = status;
m_message = message;
}
public int getExitCode() {
return m_exitCode;
}
public int getStatus() {
return m_status;
}
public String getMessage() {
return m_message;
}
private int m_exitCode;
private int m_status;
private String m_message;
}
```

The *run* method of the SchedulerJavaExtension interface is declared to throw SchedulerJobException. The following table describes the three members:

Members	Description
int m_exitCode	The framework assigns this exit code to the agent.
int m_status	The framework assigns this status code to the agent.
String m_message	The framework assigns this message to the agent.

28.8 Oracle BI Scheduler Java Extension Example

The following example illustrates how to use the previously described interfaces and class to create a custom Java action. For more information, see Section 28.1, "Using Oracle BI Scheduler Java Jobs."

This example does not contain any long running code, so it is acceptable to do nothing in the cancel method.

When the compiled class runs, it collects the ID of the user who ran the agent, the job ID of the agent, the instance ID of the agent, and all possible parameters into an output file.

```
package sample;
import java.io.*;
import java.lang.Thread;
import com.siebel.analytics.scheduler.javahostrpccalls.SchedulerJavaExtension;
import com.siebel.analytics.scheduler.javahostrpccalls.SchedulerJobException;
import com.siebel.analytics.scheduler.javahostrpccalls.SchedulerJobInfo;
/**
 *
 * @author
 */public class SimpleTest implements SchedulerJavaExtension
{
public void run(SchedulerJobInfo jobInfo) throws SchedulerJobException
{
System.out.println("JobID is:" + jobInfo.jobID());
System.out.println("Instance ID is:" + jobInfo.instanceID());
System.out.println("JobInfo to string is:" + jobInfo.toString());
try
{
File outputFile = new File("D:\\temp\\JavaJob.txt");
FileWriter out = new FileWriter(outputFile);
out.write("User ID:\t\t" + jobInfo.userID() + "\r\n");
out.write("Job ID:\t\t" + jobInfo.jobID() + "\r\n");
out.write("Instance ID:\t\t" + jobInfo.instanceID() + "\r\n");
out.write("Parameter Count:\t\t" + jobInfo.parameterCount() + "\r\n");
for(int i = 0; i < jobInfo.parameterCount(); ++i)
{
out.write("\tParameter ");
out.write(new Integer(i).toString());
out.write(":\t" + jobInfo.parameter(i) + "\r\n");
}
out.close();
}
catch(Exception ex)
{
throw new SchedulerJobException(1, 1, ex.getMessage());
}
}
public void cancel()
{
}
}
```

29

Oracle BI Scheduler Job Manager Menus

This chapter describes the Job Manager feature of the Oracle BI Administration Tool. It is the first part of the process of configuring Oracle BI Scheduler from the Job Manager.

This chapter contains the following topics:

- Section 29.1, "About the Job Manager"
- Section 29.2, "Toolbar Menus in Job Manager"
- Section 29.3, "General Oracle BI Scheduler Job Properties"
- Section 29.4, "Job Action Properties Available in Job Manager"
- Section 29.5, "Job Triggers in Job Manager"

29.1 About the Job Manager

Use the Job Manager to add, remove, modify, or cancel Oracle BI Scheduler jobs. For example, you can perform the following tasks:

- Set options for an execution schedule, such as a start time, a start date, an interval between executions, and an optional end time and date.
- Add or modify jobs using the Add Job and Modify Job dialog boxes. These dialog boxes contain three types of information:
  - General job properties
  - A script area where you can specify the actions to perform
  - A trigger area where you can specify the job trigger

    The trigger defines when the job is run.

29.2 Toolbar Menus in Job Manager

The Job Manager toolbar contains four menus, as described in the following topics:

- Section 29.2.1, "File Menu in Job Manager"
- Section 29.2.2, "Service Management Menu in Job Manager"
- Section 29.2.3, "Jobs Menu in Job Manager"
- Section 29.2.4, "Instances Menu in Job Manager"

29.2.1 File Menu in Job Manager

Table 29–1 describes the File menu options.

Table 29–1 Job Manager File Menu Options

Command	Description
Open Scheduler Connection	Opens the Machine Name dialog, which provides alternative connection mechanisms. If you run a single, non-clustered Scheduler, connect using the option Connect directly to the Active Scheduler. If the active scheduler is running on this machine (and is not configured to listen only on a particular network interface), then you might use `localhost` for the machine name and specify the port number (usually 9704). If the Scheduler is clustered, then use the option Connect Through Cluster Controllers. This option ensures that you can successfully connect irrespective of which Scheduler is currently the active Scheduler. If the system has been secured with SSL, then you must select the **SSL** check box. If the default SSL configuration is used, then you can leave all other SSL fields empty.
Close Scheduler Connection	Closes the Job Manager connection to Oracle BI Scheduler.
Exit	Shuts down the Job Manager and returns you to the Oracle BI Administration Tool. If you exit the Job Manager while a connection to Oracle BI Scheduler is still open, then the connection closes.

29.2.2 Service Management Menu in Job Manager

Table 29–2 describes the Service Management menu options.

Table 29–2 Job Manager Service Management Menu

Command	Description
Pause Scheduling	Stops all jobs from executing until scheduling is continued. Pause Scheduling is sometimes required for maintenance purposes. It allows an administrator to intervene and resolve out of control jobs. A custom job that uses excessive resources and that is being scheduled very frequently, might make any other processes on that computer ineffective, including Job Manager. Pausing scheduling offers a chance to remove or modify the job. Sometimes you might need to Pause Scheduling while Oracle BI Scheduler is stopped. In this case, scheduling continues when Oracle BI Scheduler is restarted, unless you also set the option Pause When Service Starts as described in Section 20.3.3.1, "General Scheduler Configuration Settings That Affect Agents".
Continue Scheduling	Resumes Oracle BI Scheduler's regular execution.
Stop Service	Stops the Oracle BI Scheduler service.

29.2.3 Jobs Menu in Job Manager

Table 29–3 describes the Jobs menu options.

Table 29–3 Job Manager Jobs Menu

Command	Description
Add New Job	Opens the Add New Job window, where you specify the properties for a new job.
Remove Job(s)	Removes the selected job or jobs from Oracle BI Scheduler. When a job is removed, all instances for that job are also removed.
Modify Job	Opens the Modify Job window where you can modify the properties for an existing job.

Table 29–3 (Cont.) Job Manager Jobs Menu

Command	Description
View Job	Opens the View Job window where you can view the properties for a job.
Run Job(s) Now	Immediately runs the scheduled job that you selected.
Refresh Job List	Refreshes the job information that is displayed in the Job List in the right pane. To view the instances that are associated with one or more jobs, first highlight the jobs in the job view. Then press the refresh icon in the instances view below.

29.2.4 Instances Menu in Job Manager

An Oracle BI Scheduler instance records information regarding a specific execution of a job in the Oracle BI Scheduler. Table 29–4 describes the Instances menu options.

Table 29–4 Job Manager Instances Menu

Command	Description
Cancel Instance(s)	Cancels the running job. When a job has been successfully canceled, the job's status is displayed as "Canceled."
Purge Instance	Opens the Purge Instances dialog box where you can specify the delete instance method to use. You can delete the instance by Job ID, User ID, and Before a particular time.
View Instance	Displays information about the selected instance.
Re-Run Job Instance(s) Now	Re-run a job instance. When you re-run a failed agent job, only the failed items are delivered.
Refresh Instance List	Refreshes the instance information that is displayed in the Instance List in the Job Instance View pane.

29.2.5 Instance Properties in Job Manager

Table 29–5 describes the properties of Job Manager instances.

Table 29–5 Job Manager Instance Properties

Field	Description
JobID	ID of the job that is associated with this instance.
Status: Running	This is the same for agent jobs and Script jobs. If the instance is running, then the status is running.
Status: Completed	For agent: The agent instance is set to complete if deliveries are successful to all delivery devices of the agent. For Script: This is set according to the Severity property in the script. See Table 27–2, " Oracle BI Scheduler Read/Write Script Object Properties".
Status: Failed	For agent: The agent instance is set to failed if deliveries are unsuccessful to any of the delivery devices of the agent. For Script: This is set according to the Severity property in the script. See Table 27–2, " Oracle BI Scheduler Read/Write Script Object Properties".
Status: Canceled	Canceling any instance from the Job Manager sets the status to canceled.

Table 29–5 (Cont.) Job Manager Instance Properties

Field	Description
Status: Timed Out	If the job has a maximum run time and the running time of the instance exceeds this time, then the status of the instance is set to timed out.
Status: Warning	For agent: The agent instance is set to Warning if deliveries are successful to some delivery devices but not all. For Script: This is set according to the Severity property in the script. See Table 27–2, " Oracle BI Scheduler Read/Write Script Object Properties".
InstanceID	Unique ID of this specific instance of the job.
Begin Time	The day and time that the Scheduler initiated the job instance.
End Time	The day and time that the job scheduler completed the job instance.
ExitCode: Agent	The ExitCode of an instance is set to the number of successful deliveries. The count corresponds to the number of successful deliveries to devices, and multiple devices might exist for each recipient of an agent.
ExitCode: Script	The ExitCode of an instance is set according to the ExitCode property in the script. The default is 0 (zero). See Table 27–2, " Oracle BI Scheduler Read/Write Script Object Properties"
Message	Text message that contain any error information of the instance, warnings, or general messages about the instance execution.

29.3 General Oracle BI Scheduler Job Properties

In the Add Job or Modify Job dialog box, use the fields to configure or modify the general properties for a job. Table 29–6 describes the general job properties.

In addition, Table 29–7 describes the job action properties, and Table 29–8 describes the recurrent job triggers.

Table 29–6 General Oracle BI Scheduler Job Properties

Field	Description
Name	Short, descriptive name for the job. This field is also displayed in the Job List display in the right pane of the Job Manager window.
Description	Brief description of the job that describes its actions to end users. This field is also displayed in the Job List display in the right pane of the Job Manager window.
UserID	Required for all jobs. For jobs that communicate with the Oracle BI Server or with Oracle BI Presentation Services, the UserID must be a valid Oracle Business Intelligence user ID. This field is also displayed in the Job List display in the right pane of the Job Manager window. When this job runs, Oracle BI Scheduler executes it on behalf of the user ID that is specified in this field.
Maximum Run Time MS	Specifies the maximum number of milliseconds that this job should run before it is canceled forcibly. If a job exceeds its run time, then it fails with a time-out reason code. To prevent the job from timing out, set this field to 0 (zero). **Note**: One second equals 1,000 milliseconds.

Table 29–6 (Cont.) General Oracle BI Scheduler Job Properties

Field	Description
Last Run Time	Display-only field that shows the last time that this job began execution. This field is also displayed in the Job List display in the right pane of the Job Manager window. If the date and time displayed here are for time zones where daylight savings applies, the time zone reflects the daylight savings time. For example, if (GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London, is set during the summer months, then this means BST (British Summer Time).
Next Run Time	Display-only field that shows recurrent jobs and the next time this job executes. The trigger is used to determine this value. If the date and time displayed here is for time zones where daylight savings applies, the time zone reflects the daylight savings time. For example, if (GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London, is set during the summer months, then this means BST (British Summer Time).
Running Instance Count	Display-only field that shows the number of currently running instances of this job.
Delete Job When Done	When you select this option, Oracle BI Scheduler deletes the job after its last scheduled execution as defined by its trigger. When there is no next run time, the job is done. When a job is deleted, all instances are deleted as well. For most jobs, you should not select this option, because you can delete a job manually through the Job Manager.
Disabled	When you select this option, the job script does not execute when its trigger expires. However, the next run time is still updated according to the trigger settings. The Disabled field is useful when testing or debugging a new job because an administrator can quickly disable a job without losing all information.
Execute When Missed	If you select this option while Oracle BI Scheduler is stopped (either all scheduling pauses or the Oracle BI Scheduler application stops), and if the job's next run time was missed, then the job runs after Oracle BI Scheduler restarts. If you do not select this option, then the job executes at the its next run time, as defined by its trigger.
Delete Script When Job is Removed	If you select this option, then when a job is removed, its associated job script is also removed. If many jobs reference the same job script, then this option should not be set.

29.4 Job Action Properties Available in Job Manager

Table 29–7 describes the job action properties available in the Add Job and Modify Job dialog boxes in Job Manager. Use the fields in the Script area of the Add Job or Modify Job dialog box to define the actions a job performs when it executes.

Table 29–7 Job Manager Job Action Fields

Field	Description
Script Type	Oracle BI Scheduler supports VBScript, JScript, Java, and NQCmd. Set this field according to the type of script that is referenced by the Script field. The fields that are displayed depend upon the type of script that you specify.
Load Script from File	(VBScript and JScript only) In the Job Manager, you can enter either a file name or the actual contents of a script in the Script Path field.

Table 29–7 (Cont.) Job Manager Job Action Fields

Field	Description
Script	(VBScript and JScript only) This value is either a reference to a job script file or the contents of a job script itself. If it is a reference, then enter a file name in this field, such as TestConnect.js. If no path is given, then Oracle BI Scheduler examines the directory that is referred to in the Default Script Path configuration value in Section 20.3.3.1, "General Scheduler Configuration Settings That Affect Agents". If a path is given, then the path must be accessible by the Oracle BI Scheduler application.

Parameters	(VBScript, JScript, and Java only) Field values are passed to the job script through the Parameters array. Enter one parameter per line. For example:
	`c:\oracleBI\data\scheduler`
	`cli_snowflake`
	`SELECT Lastname FROM Employee`

Class Name	(Script Type = Java) The fully qualified implementation class for the Java program action.
Class Path (jar file)	(Script Type = Java) The name of the JAR file that contains the Java program.
Additional Class Path(s)	(Script Type = Java) Other JAR files that contain the utility classes and libraries that the Java program must run properly. Include a comma-delimited list of JAR files. After this parameter, you must append a command line parameter as it is in Job Manager.
DSN (Data Source Name)	(Script Type = NQCmd) The data source name that Oracle BI Scheduler uses to connect to the BI Server. **Note**: Impersonation is used with NQCmd. The value in the User ID is the user that Oracle BI Scheduler tries to impersonate when connecting to the BI Server.
SQL Input File	(Script Type = NQCmd) The fully qualified path to the SQL file that NQCmd executes. Type the full path or click the ... button to browse to the file's location. This field is typically used for files that are generated by the aggregate persistence feature.

Additional Command Line Parameters	(Script Type = NQCmd) Parameters that are passed to NQCmd. Enter one parameter per line. For example:
	`-o`
	`D:\foo\bar.txt`

29.5 Job Triggers in Job Manager

A job trigger determines when and how often a job executes. Use the fields in the Trigger area of the Add Job or Modify Job dialog box to define the actions that a job performs when it executes.

There are two types of Oracle BI Scheduler Job triggers—*single-run* triggers and *recurrent* triggers.

29.5.1 Single-Run Triggers

Use the Trigger Type list to select the trigger type. Single-run triggers perform the action once. There are two single-run triggers:

- **Run Now.** This trigger specifies that the job runs immediately. It executes only one time.

- **Run Once.** Jobs of this trigger type execute at the date and time that is specified in the Begin Date and Start Time fields, which become active when you selected **Run Once**. An error occurs if the given time is in the past. If you select the **Set Start Time To Now** option, then this trigger is equivalent to the **Run Now** trigger.

29.5.2 Recurrent Triggers

All recurrent triggers specify that the job execute over a period of time at given intervals. Fields used by recurrent triggers are described in Table 29–8. Recurrent Trigger Types are described in Table 29–9.

Table 29–8 Job Manager Recurrent Trigger Fields

Field	Description
Timezone	Specifies the time zone that is used to execute the job. Displays the default time zone as specified in the instanceconfig.xml file. For information, see Section 15.5, "Description of Time Zone Settings.")
Begin Date	Specifies the date when the first recurrent interval runs. The recurrent interval is defined as the time between Start Time and End Time. This field is hidden if you select the Set Start Time to Now option.
End Date	Specifies the date when the last recurrent interval is run. Becomes active when the Has End Date option is set. If no end date is set, then the job runs forever.
Start Time	Specifies the lower bounds of the recurrent interval. The job's first execution for a day occurs at the time that is specified in this value.
End Time	Specifies the upper bounds of the recurrent interval. The job's last execution for a given day occurs at or before the time that is specified in this value. If this value is less than the Start Time value, then the interval spans midnight of the given day. For example, a trigger with a start time of 11:00 P.M. and an End Time of 2:00 A.M. starts its execution on the date that is specified in Begin Date at 11:00 P.M. and continues until 2:00 A.M. on the following day.
Has End Date	If you select this option, then specify an End Date. If you do not select this option, then the job stays scheduled. **Note**: The schedule is perpetual; the job instance is not. If you restart Oracle BI Scheduler, then the next run time is set as dictated by the job schedule. If an instance is running while you stop Oracle BI Scheduler, then it is canceled.
Set Start Time To Now	If you select this option, then the Begin Date and Start Time fields are ignored and their values are populated with Oracle BI Scheduler's current date and time.
Interval in Minutes	Specifies the number of minutes between subsequent executions of a job during the recurrent interval. A job starts execution promptly at its Start Time and executes again every *n* minutes, where *n* is the value of this field.
Maximum Concurrent Instances	If a job executes every *n* minutes (from the Interval in Minutes field), then a long-running job might have overlapping executions. Use this field to set the number of concurrent running instances. For an unlimited number of concurrent instances, set this value to zero.

29.5.2.1 Recurrent Trigger Types

The recurrent trigger types that are available from the Trigger Type list are described in Table 29–9. The fields pertain to all recurrent triggers. Depending on the trigger type that you select, additional options become active. The examples in the table illustrate how these additional options can be used.

Table 29–9 Job Manager Recurrent Trigger Types

Trigger Type	Description and Example
Daily	Runs a job every day or every few days. The Days Interval field specifies the number of days between each subsequent recurrent interval. For example: To run a job every hour between 8:00 A.M. and 5:00 P.M. starting on January 1, 2010 and ending on January 15, 2010, set the Begin Date to 1/1/10, the Start Time to 8:00 A.M., and the End Time to 5:00 P.M. Set the Has End Date flag, the End Date to 1/15/10, the Interval in Minutes to 60, and the Days Interval to 1. To run a job every five minutes forever, set the Begin Date to the desired date, the Start Time to 12:00 P.M., the end time to 11:59 A.M., the Interval In Minutes to 5, and the Days Interval to 1.
Weekly	Runs a job on specified days of the week. The Weeks Interval specifies the number of weeks between each execution. The Days of the Week field specifies on which days the execution occurs. For example: To run a job at noon every other week on Mondays, Wednesdays, and Fridays, set the Begin Date to the desired date, the Start Time and End Time to 12:00 P.M., the Interval in Minutes to 1, the Weeks Interval to 2, and the Days of the Week to Monday, Wednesday, and Friday.
Monthly by Date	Runs a job on specific days of the month. The Months field specifies in which months this job executes. The Days field specifies which days of those months. If the given day does not exist for a given month, then that day is ignored. For example: To run a job at 5:00 P.M. on the 1st and 15th of January, February, and March, set the Begin Date to January 1, the Start Time and End Time to 5:00 P.M., the Interval in Minutes to 1, the Months to January, February, and March, and the Days to 1 and 15. To run a job at 2:00 A.M. on every leap day (February 29th), set the Begin Date to January 1, the Start Time and End Time to 2:00 A.M., the Interval in Minutes to 1, the Months to February, and the Days to 29.

Table 29–9 (Cont.) Job Manager Recurrent Trigger Types

Trigger Type	Description and Example
Monthly by DOW (Day of Week)	Runs a job on specific occurrences of specified days of the week during given months. The Months field specifies which months this job executes. The Days of the Week field specifies which days of the week the job executes during those months. The Occurrence field specifies which of those days to execute. The occurrence can be any or all of First, Second, Third, Fourth, and Last. The Last value specifies that either the fourth or fifth occurrence of a given day is used, depending on whether there are four or five occurrences during that month. For example: To run a job on the first and third Fridays of December every hour between the hours of 4:00 A.M. and 8:00 P.M., set the Begin Date to the desired date, the Start Time to 4:00 A.M., the End Time to 8:00 P.M., the Interval in Minutes to 60, the Months to December, the Days of the Week to Friday, and the Occurrence to the First and the Third. To run a job at 3:00 A.M. every time that Daylight Saving Time switches over to Standard Time, set the Begin Date to the desired date, the Start Time and End Time to 3:00 A.M., the Months to October, the Days of the Week to Sunday, and the Occurrence to Last.

Part VIII

Automating Management of Oracle Business Intelligence

This part explains how to use the Oracle BI Systems Management API. It contains the following chapter:

30

Introducing the Oracle BI Systems Management API

This chapter describes the Oracle BI Systems Management Application Programming Interface (API) and contains the following topics:

- Section 30.1, "What is the Oracle BI Systems Management API?"
- Section 30.2, "Accessing Oracle BI Systems Management API Methods"
- Section 30.3, "Using the Oracle BI Systems Management API to Manage and Configure Oracle Business Intelligence"
- Section 30.4, "Descriptions of JMX Admin MBeans and Methods for Oracle BI Systems Management API"

30.1 What is the Oracle BI Systems Management API?

The Oracle BI Systems Management API is a programming interface that provides access to Oracle Business Intelligence JMX Admin MBeans, enabling a developer to programmatically perform system administration tasks for Oracle Business Intelligence. The system administration tasks that can be performed using the Oracle BI Systems Management API are also available using Fusion Middleware Control.

The Oracle BI Systems Management API centralizes the management of system configuration by automatically propagating changes to all of the files that require updates in multiple locations.

Alternative methods for modifying system configuration settings for Oracle Business Intelligence are available. For information, see Section 3.1, "Configuring Oracle Business Intelligence."

The Oracle BI Systems Management API and its Java methods are described in a JavaDoc file that is located in the following directory:

ORACLE_HOME/doc/javadoc/bifoundation/jmxapi/index.html

> **Note:** You cannot display the Log Viewer or Performance Metrics using the Oracle BI Systems Management API.

30.2 Accessing Oracle BI Systems Management API Methods

This section describes the following ways to access Oracle BI Systems Management API methods:

- Using an Oracle WebLogic Server Scripting Tool (WLST) command (interactive mode)

  You can use the command line to enter a command and view the response at a command-line prompt. This method is useful for learning the tool, prototyping command syntax, and verifying configuration options before building a script. Using WLST interactively is particularly useful for getting immediate feedback after making a critical configuration change. The WLST scripting shell maintains a persistent connection with an instance of WebLogic Server.

- Using a WLST command in a script (Script mode)

  You can use a script to invoke a sequence of WLST commands without requiring your input, much like a shell script. Scripts contain WLST commands in a text file with a .py file extension, such as filename.py. You use script files with the Jython commands for running scripts. The Jython scripting language is used because it is freely available and compatible with the Java platform.

- Using a Java program

  You can use a Java program to contact the JMX server directly.

For information about the Oracle BI Systems Management API methods, see Section 30.4, "Descriptions of JMX Admin MBeans and Methods for Oracle BI Systems Management API."

30.3 Using the Oracle BI Systems Management API to Manage and Configure Oracle Business Intelligence

This section describes using the Oracle BI Systems Management API to manage and configure Oracle Business Intelligence, and contains the following topics:

- Section 30.3.1, "Displaying Oracle Business Intelligence Status Using the Oracle BI Systems Management API"
- Section 30.3.2, "Starting and Stopping Oracle Business Intelligence Using the Oracle BI Systems Management API"
- Section 30.3.3, "Updating Configuration Settings Using the Oracle BI Systems Management API"
- Section 30.3.4, "Scaling Out for High Availability and Performance Using the Oracle BI Systems Management API"

30.3.1 Displaying Oracle Business Intelligence Status Using the Oracle BI Systems Management API

You can use the Oracle BI Systems Management API to display the status of Oracle Business Intelligence using either WLST commands or a Java program as described in the following sections:

- Section 30.3.1.1, "Example: Using WLST Commands in a Script to Display the Status of Oracle Business Intelligence"
- Section 30.3.1.2, "Example: Using a Java Program to Display the Status of Oracle Business Intelligence"

30.3.1.1 Example: Using WLST Commands in a Script to Display the Status of Oracle Business Intelligence

This example connects a user to Oracle Business Intelligence and displays the service status.

To use WLST commands in a script to display the status of Oracle Business Intelligence:

1. Open a text editor and create a WLST script similar to the following example:

```
# Example to demonstrate connecting to a BIEE domain using WLST
#
# This scripts expects the following arguments:
#
# 1. wls.host (localhost)
# 2. wls.port (7001)
# 3. wls.user  (user1)
# 4. wls.password  (password1)
# ===================================================================

import sys
import os

# Check the arguments to this script are as expected.
# argv[0] is script name.
argLen = len(sys.argv)
if argLen -1 != 4:
    print "ERROR: got ", argLen -1, " args."
    print "USAGE: wlst.cmd wls_connect.py WLS_HOST WLS_PORT WLS_USER WLS_
PASSWORD"
    print "   eg: wlst.cmd wls_connect.py localhost 7001 user1 password1"
    exit()

WLS_HOST = sys.argv[1]
WLS_PORT = sys.argv[2]
WLS_USER = sys.argv[3]
WLS_PW = sys.argv[4]

print 'Connecting to '+ WLS_HOST+ ':' + WLS_PORT + ' as user: ' + WLS_USER + '
...'

# Connect to WLS
connect(WLS_USER, WLS_PW, WLS_HOST+ ':' + WLS_PORT);

print 'Connecting to Domain ...'
domainCustom()
cd ('oracle.biee.admin')
print 'Connecting to BIDomain MBean ...'
cd ('oracle.biee.admin:type=BIDomain,group=Service')

biinstances = get('BIInstances')
biinstance = biinstances[0]

print 'Connecting to BIInstance MBean ...'
cd ('..')
cd (biinstance.toString())

servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus

exit()
```

2. Save the script.

3. Navigate to the folder \MW_HOME\wlserver_10.3\common\bin and run the script, entering any arguments required.

   For example, you might enter the following at the command line:

   `./wlst.sh wlst_connect.py localhost 7001 user1 password1` (UNIX)

   `wlst wlst_connect.cmd localhost 7001 user1 password1` (Windows)

   The command window (and log file), displays prompts and messages, for example:

   ```
   Initializing WebLogic Scripting Tool (WLST) ...

   Welcome to WebLogic Server Administration Scripting Shell

   Type help() for help on available commands

   Connecting to localhost:7001 as user: user1 ...
   Connecting to t3://localhost:7001 with userid password1 ...
   Successfully connected to Administration Server 'AdminServer' that belongs to
   domain 'bifoundation_domain'.

   Warning: An insecure protocol was used to connect to the
   server. To ensure on-the-wire security, the SSL port or
   Admin port should be used instead.

   Connecting to Domain ...
   Location changed to domain custom tree.
   This is a writable tree with No root.
   For more help, use help(domainCustom)

   Connecting to BIDomain MBean ...
   Connecting to BIInstance MBean ...
   BIInstance MBean; ServiceStatus: FULLY_STARTED


   Exiting WebLogic Scripting Tool.
   ```

For more information about using WLST commands, see the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

30.3.1.2 Example: Using a Java Program to Display the Status of Oracle Business Intelligence

This example connects to Oracle Business Intelligence and displays the current status. The Java program in this example also contains additional methods (commented out) which you can use to start and stop Oracle Business Intelligence. For more information, see Section 30.3.2.2, "Example: Using a Java Program to Stop and Start Oracle Business Intelligence."

To use a Java program with the Systems management API to display the status of Oracle Business Intelligence:

1. Open a Java editor and compile a Java program similar to the following example:

   ```
   package com.oracle.bi.example;
   import java.util.Hashtable;
   import javax.management.MBeanServerConnection;
   ```

```
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;
import javax.naming.Context;

/**
 * Example class showing how to connect to the Oracle BIEE AdminMBeans JMX
Interface.
 * <br/>
 * This example is intentionally succinct for clarity, so for example,
 * it does not do any exception handling.
 * Any production code should include robust error handling.
 * Note that you need <code>wljmxclient.jar</code> and
<code>wlclient.jar</code> on the classpath.
 */
public class MBeanAPIExample1 {
       private String host;
       private int port;
       private String username;
       private String password;
       private MBeanServerConnection mbs = null;
       private ObjectName biDomainMBeanName;
       private ObjectName biInstanceMBeanName;
       private MBeanAPIExample1(String[] args) {
               if (args.length < 4) {
                        System.out.println("Usage: MBeanAPIExample1 host port
                         username password");
                         throw new IllegalArgumentException("Not enough
                          arguments");
             }
             host = args[0];
             port = Integer.parseInt(args[1]);
             username = args[2];
             password = args[3];
       }

       private void connect() throws Exception {
             String jmxUrl = "service:jmx:t3://" + host + ":" + port +
              "/jndi/weblogic.management.mbeanservers.domainruntime";
             System.out.println("Connecting using URL: " + jmxUrl + " ...");

             Hashtable<String,String> h = new Hashtable<String,String>();
             h.put(Context.SECURITY_PRINCIPAL,   username);
             h.put(Context.SECURITY_CREDENTIALS, password);
             h.put(JMXConnectorFactory.PROTOCOL_PROVIDER_PACKAGES,
              "weblogic.management.remote");
             JMXConnector jmxConnector = JMXConnectorFactory.connect(new
               JMXServiceURL(jmxUrl), h);
             mbs = jmxConnector.getMBeanServerConnection();
             System.out.println("Connected OK");
             biDomainMBeanName = new
               ObjectName("oracle.biee.admin:type=BIDomain,group=Service");
             ObjectName[]biInstanceMBeanNames=(ObjectName[])mbs.getAttribute
               (biDomainMBeanName, "BIInstances");
             biInstanceMBeanName = biInstanceMBeanNames[0];
                // there should only be one
             System.out.println("Found BI Instance MBean: " +
                 biInstanceMBeanName);
       }
       private void displayStatus() throws Exception {
             String status = (String)mbs.getAttribute(biInstanceMBeanName,
                "ServiceStatus");
             System.out.println("Status is: " + status);
       }
       private void stopAll() throws Exception {
             System.out.println("Stopping instance ...");
             Object[] args = new Object[]{};
             String[] sig = new String[]{};
             mbs.invoke(biInstanceMBeanName, "stop", args, sig);
             System.out.println("Stopped instance OK");
       }
       private void startAll() throws Exception {
            System.out.println("Starting instance ...");
            Object[] args = new Object[]{};
            String[] sig = new String[]{};
            mbs.invoke(biInstanceMBeanName, "start", args, sig);
            System.out.println("Started instance OK");
       }
       /**
        * @param args
        */
       public static void main(String[] args) throws Exception {
            MBeanAPIExample1 example = new MBeanAPIExample1(args);
            example.connect();
            example.displayStatus();
//          example.stopAll();
//          example.startAll();
       }
}
```

2. Navigate to the \<wlserver>\server\lib folder and run the Java program, providing the class path, class name and any arguments required.

   java -classpath *<JavaClassName>* *<host>* *<port>* *<username>* *<password>*

   For example:

   java -classpath <*MW_HOME*>wls_10.3/server/lib/wjmxclient.jar com.oracle.bi.example.MBeanAPIExample1 localhost 7001 user1 password1

   The command window (and log file) displays prompts and messages, for example:

```
Connecting using URL:
service:jmx:t3://localhost:7001/jndi/weblogic.management.mbeanservers.domainrun
time
...
Connected OK
Found BI Instance MBean:
oracle.biee.admin:type=BIDomain.BIInstance,biInstance=coreapplication,group=Ser
vice

Status is: FULLY_STARTED
```

30.3.2 Starting and Stopping Oracle Business Intelligence Using the Oracle BI Systems Management API

You can use the Oracle BI Systems Management API start and stop Oracle Business Intelligence using either WLST commands or a Java program as described in the following sections:

- Section 30.3.2.1, "Example: Using WLST Commands to Stop and Start Oracle Business Intelligence"
- Section 30.3.2.2, "Example: Using a Java Program to Stop and Start Oracle Business Intelligence"

30.3.2.1 Example: Using WLST Commands to Stop and Start Oracle Business Intelligence

This example stops and starts the Oracle Business Intelligence instance, and displays the status during each stage of the process.

To use WLST commands to stop and start Oracle Business Intelligence:

1. Open a text editor and create a WLST script similar to the following:

```
# Example to demonstrate connecting to a BIEE domain using WLST and stopping
and restarting the instance
#
# This scripts expects the following arguments:
#
# 1. wls.host (localhost)
# 2. wls.port (7001)
# 3. wls.user  (user1)
# 4. wls.password  (password1)
# ===================================================================
import sys
import os
# Check the arguments to this script are as expected.
# argv[0] is script name.
argLen = len(sys.argv)
if argLen -1 != 4:
    print "ERROR: got ", argLen -1, " args."
    print "USAGE: wlst_stop_start.cmd wls_stop_start_obi.py WLS_HOST WLS_PORT
WLS_USER WLS_PASSWORD"
    print "   eg: wlst_stop_start.cmd wls_stop_start_obi.py localhost 7001
user1 password1"
    exit()
WLS_HOST = sys.argv[1]
WLS_PORT = sys.argv[2]
WLS_USER = sys.argv[3]
WLS_PW = sys.argv[4]
print 'Connecting to '+ WLS_HOST+ ':' + WLS_PORT + ' as user: ' + WLS_USER + '
...'
# Connect to WLS
connect(WLS_USER, WLS_PW, WLS_HOST+ ':' + WLS_PORT);
print 'Connecting to Domain ...'
domainCustom()
cd ('oracle.biee.admin')
print 'Connecting to BIDomain MBean ...'
cd ('oracle.biee.admin:type=BIDomain,group=Service')
biinstances = get('BIInstances')
biinstance = biinstances[0]
print 'Connecting to BIInstance MBean ...'
```

```
cd ('..')
cd (biinstance.toString())
servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus
print 'Calling stop ...'
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('stop', objs, strs)
servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus
print 'Calling start ...'
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('start', objs, strs)
servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus
exit()
```

2. Save the script.
3. Navigate to the folder \MW_HOME\\wlserver_10.3\\common\\bin and run the script, entering any arguments required.

   For example, you might type one the following at the command line:

   `./wlst.sh wlst_stop_start_obi.py localhost 7001 user1 password1` (UNIX)

   `wlst wlst_wlst_stop_start_obi.cmd localhost 7001 user1 password1` (Windows)

   The command window (and log file), displays prompts and messages, for example:

```
Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Connecting to localhost:7001 as user: user1 ...
Connecting to t3://localhost:7001 with userid password1 ...
Successfully connected to Administration Server 'AdminServer' that belongs to
domain 'bifoundation_domain'.

Warning: An insecure protocol was used to connect to the
server. To ensure on-the-wire security, the SSL port or
Admin port should be used instead.

Connecting to Domain ...
Location changed to domain custom tree.
This is a writable tree with No root.
For more help, use help(domainCustom)

Connecting to BIDomain MBean ...
Connecting to BIInstance MBean ...
BIInstance MBean; ServiceStatus: FULLY_STARTED
Calling stop ...
BIInstance MBean; ServiceStatus: STOPPED
Calling start ...
BIInstance MBean; ServiceStatus: FULLY_STARTED
```

```
Exiting WebLogic Scripting Tool.
```

30.3.2.2 Example: Using a Java Program to Stop and Start Oracle Business Intelligence

Refer to the example in Section 30.3.1.2, "Example: Using a Java Program to Display the Status of Oracle Business Intelligence," remove the comments (/) before whichever method you want to run (example.stopAll(); or example.startAll();), as shown in the following main method:

```
       /**
        * @param args
        */
       public static void main(String[] args) throws Exception {
            MBeanAPIExample1 example = new MBeanAPIExample1(args);
            example.connect();
            example.displayStatus();
/**         example.stopAll();
*/
/**         example.startAll();
*/
       }
```

30.3.3 Updating Configuration Settings Using the Oracle BI Systems Management API

You can use the Oracle BI Systems Management API to change Oracle Business Intelligence configuration settings using WLST commands or a Java program as follows:

- Section 30.3.3.1, "Example: Using WLST to Edit the E-Mail Sender Display Name Configuration Setting"
- Section 30.3.3.2, "Example: Using a Java Program to Edit the E-Mail Sender Display Name Configuration Setting"

You can copy the examples in this section, replacing MBean details with details of other MBeans to update different Oracle Business Intelligence configuration settings (for example, log file settings, security settings). For more information, see Section 30.4, "Descriptions of JMX Admin MBeans and Methods for Oracle BI Systems Management API."

30.3.3.1 Example: Using WLST to Edit the E-Mail Sender Display Name Configuration Setting

This example connects to Oracle Business Intelligence, invokes the system lock, updates the value of the E-Mail Sender Display Name configuration setting, and releases the system lock.

To use WLST to update the E-Mail Sender Display Name configuration setting:

1. Open a text editor and enter WLST commands similar to the following:

```
# Example to demonstrate connecting to a BIEE domain using WLST,
# and changing a simple configuration setting
#
# This script expects the following arguments:
#
# 1. wls.host (localhost)
# 2. wls.port (7001)
# 3. wls.user  (user1)
# 4. wls.password  (password1)
# 5. new email displayname
# ===================================================================

import sys
import os

# Check the arguments to this script are as expected.
# argv[0] is script name.
argLen = len(sys.argv)
if argLen -1 != 5:
    print "ERROR: got ", argLen -1, " args."
    print "USAGE: wlst.cmd wls_connect.py WLS_HOST WLS_PORT WLS_USER WLS_
PASSWORD EmailDisplayName"
    print "   eg: wlst.cmd wls_connect.py localhost 7001 user1 password1
FredBloggs"
    exit()

WLS_HOST = sys.argv[1]
WLS_PORT = sys.argv[2]
WLS_USER = sys.argv[3]
WLS_PW = sys.argv[4]
newDisplayName = sys.argv[5]

print 'Connecting to '+ WLS_HOST+ ':' + WLS_PORT + ' as user: ' + WLS_USER + '
...'

# Connect to WLS
connect(WLS_USER, WLS_PW, WLS_HOST+ ':' + WLS_PORT);

print 'Connecting to Domain ...'
domainCustom()
cd ('oracle.biee.admin')
print 'Connecting to BIDomain MBean ...'
cd ('oracle.biee.admin:type=BIDomain,group=Service')
#bidomain=cmo

print 'Calling lock ...'
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('lock', objs, strs)

biinstances = get('BIInstances')
biinstance = biinstances[0]

print 'Connecting to BIInstance MBean ...'
cd ('..')
cd (biinstance.toString())

servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus

biemailbean = get('EmailConfiguration')
print 'Connecting to BIInstance Email MBean ...'
cd ('..')
cd (biemailbean.toString())

oldDisplayName=get('SenderDisplayName')
```

```
print 'Existing email displayname is: ' + oldDisplayName
print 'Changing email displayname to: ' + newDisplayName + ' ...'
set('SenderDisplayName', newDisplayName)

print 'Calling commit ...'
cd ('..')
cd ('oracle.biee.admin:type=BIDomain,group=Service')
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('commit', objs, strs)

print 'Committed OK'

exit()
```

2. Save the script.
3. Navigate to the folder *MW_HOME*\wlserver_10.3\common\bin and run the script, entering any arguments required.

   For example, you might enter the following at the command line:

   `./wlst.sh wlst_email_disp_name_obi.py localhost 7001 user1 password1` (UNIX)

   `wlst wlst_wlst_email_disp_name_obi.cmd localhost 7001 user1 password1` (Windows)

   The command window (and log file) displays prompts and messages, for example:

```
Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Connecting to localhost:7001 as user: user1 ...
Connecting to t3://localhost:7001 with userid password1 ...
Successfully connected to Administration Server 'AdminServer' that belongs to
domain 'bifoundation_domain'.

Warning: An insecure protocol was used to connect to the
server. To ensure on-the-wire security, the SSL port or
Admin port should be used instead.

Connecting to Domain ...
Location changed to domain custom tree.
This is a writable tree with No root.
For more help, use help(domainCustom)

Connecting to BIDomain MBean ...
Connecting to BIInstance MBean ...
BIInstance MBean; ServiceStatus: FULLY_STARTED
Connecting to BIInstance Email MBean ...
Existing email displayname is: Oracle Business Intelligence
Changing email displayname to: FredBloggs ...
Calling commit ...
Committed OK

Exiting WebLogic Scripting Tool.
```

30.3.3.2 Example: Using a Java Program to Edit the E-Mail Sender Display Name Configuration Setting

To use a Java program to Edit the E-Mail Sender Display Name Configuration Setting:

1. Open a Java editor and compile a Java program similar to the following example:

```
package com.oracle.bi.example;
import java.util.Hashtable;
import javax.management.Attribute;
import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;
import javax.naming.Context;

/**
 * Example class showing how to connect to the Oracle BIEE AdminMBeans JMX
Interface and change a simple configuration setting..
 * <br/>
 * This example is intentionally succinct for clarity, so for example,
 * it does not do any exception handling.
 * Any production code should include robust error handling.
 *
 * Note that you need <code>wljmxclient.jar</code> and
<code>wlclient.jar</code> on the classpath.
 */
public class MBeanAPIExample3Edit {
       private String host;
       private int port;
       private String username;
       private String password;
       private String newDisplayName;
       private MBeanServerConnection mbs = null;
       private ObjectName biDomainMBeanName;
       private ObjectName biInstanceMBeanName;
       private ObjectName biEmailMBeanName;
       private MBeanAPIExample3Edit(String[] args) {
               if (args.length < 5) {
                 System.out.println("Usage: MBeanAPIExample3Edit host port
username password newDisplayName");
                throw new IllegalArgumentException("Not enough arguments");
             }
             host = args[0];
             port = Integer.parseInt(args[1]);
             username = args[2];
             password = args[3];
             newDisplayName = args[4];
       }
       private void connect() throws Exception {
             String jmxUrl = "service:jmx:t3://" + host + ":" + port +
              "/jndi/weblogic.management.mbeanservers.domainruntime";
             System.out.println("Connecting using URL: " + jmxUrl + " ...");
             Hashtable<String,String> h = new Hashtable<String,String>();
             h.put(Context.SECURITY_PRINCIPAL,   username);
             h.put(Context.SECURITY_CREDENTIALS, password);
             h.put(JMXConnectorFactory.PROTOCOL_PROVIDER_PACKAGES,
              "weblogic.management.remote");
             JMXConnector jmxConnector = JMXConnectorFactory.connect(new
               JMXServiceURL(jmxUrl), h);
             mbs = jmxConnector.getMBeanServerConnection();
             System.out.println("Connected OK");

             biDomainMBeanName = new
               ObjectName("oracle.biee.admin:type=BIDomain,group=Service");
             ObjectName[]biInstanceMBeanNames=(ObjectName[])mbs.getAttribute
               (biDomainMBeanName, "BIInstances");
             biInstanceMBeanName = biInstanceMBeanNames[0];
                // there should only be one
             System.out.println("Found BI Instance MBean: " +
                 biInstanceMBeanName);
       }
       private void lock() throws Exception{
             System.out.println("Locking domain ...");
             Object[] args = new Object[]{};
             String[] sig  = new String[]{};
             mbs.invoke(biDomainMBeanName, "lock", args, sig);
             System.out.println("Locked domain OK");
       }
       private void commit() throws Exception{
             System.out.println("Committing changes ...");
             Object[] args = new Object[]{};
             String[] sig  = new String[]{};
             mbs.invoke(biDomainMBeanName, "commit", args, sig);
             System.out.println("Committed changes OK");
       }
       private void getEmailBean() throws Exception{
             System.out.println("Getting Email MBean ...");
             biEmailMBeanName =
(ObjectName)mbs.getAttribute(biInstanceMBeanName, "EmailConfiguration");
             System.out.println("Found BI Email MBean: " + biEmailMBeanName);
       }
        private void updateEmailDisplayName() throws Exception {
            String oldDisplayName = (String)mbs.getAttribute(biEmailMBeanName,
"SenderDisplayName");
            System.out.println("Existing email displayname is: " +
oldDisplayName);
            System.out.println("Changing email displayname to: " +
newDisplayName);
            mbs.setAttribute(biEmailMBeanName, new
Attribute("SenderDisplayName", newDisplayName));
       }
       /**
        * @param args
        */
       public static void main(String[] args) throws Exception {
            MBeanAPIExample3Edit example = new MBeanAPIExample3Edit(args);
            example.connect();
            example.getEmailBean();
            example.lock();
            example.updateEmailDisplayName();
            example.commit();
       }
}
```

2. Navigate to the \<wlserver>\server\lib folder and run the Java program giving the class path, class name and any arguments required.

java -classpath <*JavaClassName*> <*host*> <*port*> <*username*> <*password*> <*SenderDisplayName*>

For example:

java -classpath <*MW_HOME*>wls_10.3/server/lib/wjmxclient.jar com.oracle.bi.example.MBeanAPIExample3Edit localhost 7001 user1 password1 FredBloggs

Use quotation marks to include spaces, for example "Fred Bloggs".

The command window (and log file), displays prompts and messages, for example:

```
Connecting using URL:
service:jmx:t3://localhost:7001/jndi/weblogic.management.mbeanservers.domainrun
time ...
Connected OK
Found BI Instance MBean:
oracle.biee.admin:type=BIDomain.BIInstance,biInstance=coreapplication,group=Ser
vice
Getting Email MBean ...
Found BI Email MBean:
oracle.biee.admin:type=BIDomain.BIInstance.EmailConfiguration,biInstance=coreap
plication,group=Service
Locking domain ...
Locked domain OK
Existing email displayname is: Oracle Business Intelligence
Changing email displayname to: FredBloggs
Committing changes ...
Committed changes OK
```

30.3.4 Scaling Out for High Availability and Performance Using the Oracle BI Systems Management API

You can use the Oracle BI Systems Management API to scale out to additional host computers for high availability and performance. Use the Enterprise Install to install Oracle Business Intelligence onto a new host computer, and then scale out using steps similar to those described in the following examples:

- Section 30.3.4.1, "Example: Using WLST to Scale Out to Servers"
- Section 30.3.4.2, "Example: Using a Java Program to Scale Out to Servers"

30.3.4.1 Example: Using WLST to Scale Out to Servers

This example connects a user to Oracle Business Intelligence and scales out to a server.

To use WLST commands in a script to scale out to servers:

1. Open a text editor and create a WLST script similar to the following example:

```
# Example to demonstrate connecting to a BIEE domain using WLST
#  and do some scale-out operations.
#
# This scripts expects the following arguments:
#
# 1. wls.host (localhost)
# 2. wls.port (7001)
# 3. wls.user  (user1)
# 4. wls.password  (password1)
# 5. oracle instance name
# 6. new count of bi servers
# ===================================================================

import sys
import os

# Check the arguments to this script are as expected.
# argv[0] is script name.
argLen = len(sys.argv)
if argLen -1 != 6:
    print "ERROR: got ", argLen -1, " args."
    print "USAGE: wlst.cmd wls_connect.py WLS_HOST WLS_PORT WLS_USER WLS_
PASSWORD"  OracleInstance numBIServers
    print "   eg: wlst.cmd wls_connect.py localhost 7001 user1 password1
instance2 2"
    exit()

WLS_HOST = sys.argv[1]
WLS_PORT = sys.argv[2]
WLS_USER = sys.argv[3]
WLS_PW = sys.argv[4]
oracleInstanceName = sys.argv[5]
# convert string to int which we need later
newNumBIServers = int(sys.argv[6])


print 'Connecting to '+ WLS_HOST+ ':' + WLS_PORT + ' as user: ' + WLS_USER + '
...'

# Connect to WLS
connect(WLS_USER, WLS_PW, WLS_HOST+ ':' + WLS_PORT);

print 'Connecting to Domain ...'
domainCustom()
cd ('oracle.biee.admin')
print 'Connecting to BIDomain MBean ...'
cd ('oracle.biee.admin:type=BIDomain,group=Service')

biinstances = get('BIInstances')
biinstance = biinstances[0]

oracleinstances = get('OracleInstances')

print 'Connecting to BIInstance MBean ...'
cd ('..')
print 'About to cd to: ' + biinstance.toString()
cd (biinstance.toString())

servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus

print 'Looping through OracleInstances looking for: ' + oracleInstanceName
matching = false
for oi in oracleinstances:
    cd('..')
    print 'About to cd to: ' + oi.toString()
    cd ( oi.toString() )
    id = get('Id')
    print 'Found instance with id: ' + id
    if id == oracleInstanceName:
        matching = oi
        break

if matching :
    print 'Found matching OracleInstance...'
else :
    print 'Failed to find target OracleInstance with name: ' +
oracleInstanceName
    exit()

deployments = get('BIInstanceDeployments')
# Should only ever be one of these
deployment = deployments[0]
print 'Acquired BIInstanceDeployment : ' + deployment.toString()
cd('..')
cd ('oracle.biee.admin:type=BIDomain,group=Service')

print 'Calling lock ...'
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('lock', objs, strs)

cd('..')
cd(deployment.toString())
numBIServers = get('numBIServers')
print 'Current number of BI Servers is:', numBIServers
print 'About to set number of BI Servers to:', newNumBIServers
set('numBIServers', newNumBIServers )

print 'Successfully set number of BI Servers to', newNumBIServers
print 'Calling commit ...'
cd ('..')
cd ('oracle.biee.admin:type=BIDomain,group=Service')
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('commit', objs, strs)

print 'Committed OK'

print 'Connecting to BIInstance MBean ...'
cd ('..')
cd (biinstance.toString())

servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus

print 'Calling stop ...'
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('stop', objs, strs)

servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus

print 'Calling start ...'
objs = jarray.array([], java.lang.Object)
strs = jarray.array([], java.lang.String)
invoke('start', objs, strs)
```

```
servicestatus=get('ServiceStatus')
print 'BIInstance MBean; ServiceStatus: ' + servicestatus


exit()
```

2. Save the script.
3. Navigate to the folder \MW_HOME\\wlserver_10.3\\common\\bin and run the script, entering any arguments required.

   For example, you might enter the following at the command line:

   `./wlst.sh wlst_scaleout_obi.py localhost 7001 user1 password1` (UNIX)

   `wlst wlst_wlst_scaleout_obi.cmd localhost 7001 user1 password1` (Windows)

   The command window (and log file), displays prompts and messages, for example:

```
Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Connecting to localhost:7001 as user: user1 ...
Connecting to t3://localhost:7001 with userid password1 ...
Successfully connected to Administration Server 'AdminServer' that belongs to
domain 'bifoundation_domain'.

Warning: An insecure protocol was used to connect to the
server. To ensure on-the-wire security, the SSL port or
Admin port should be used instead.

Connecting to Domain ...
Location changed to domain custom tree.
This is a writable tree with No root.
For more help, use help(domainCustom)

Connecting to BIDomain MBean ...
Connecting to BIInstance MBean ...
About to cd to:
oracle.biee.admin:type=BIDomain.BIInstance,biInstance=coreapplication,group=Ser
vice
BIInstance MBean; ServiceStatus: FULLY_STARTED
Looping through OracleInstances looking for: instance1
About to cd to:
oracle.biee.admin:oracleInstance=instance1,type=BIDomain.OracleInstance,group=S
ervice
Found instance with id: instance1
Found matching OracleInstance...
Acquired BIInstanceDeployment :
oracle.biee.admin:oracleInstance=instance1,type=BIDomain.OracleInstance.BIInsta
nceDeployment,biInstance=coreapplication,group=Service
Calling lock ...
Current number of BI Servers is: 1
About to set number of BI Servers to: 6
Successfully set number of BI Servers to 6
Calling commit ...
```

```
Committed OK
Connecting to BIInstance MBean ...
BIInstance MBean; ServiceStatus: PARTIALLY_STARTED
Calling stop ...
BIInstance MBean; ServiceStatus: PARTIALLY_STARTED
Calling start ...
BIInstance MBean; ServiceStatus: FULLY_STARTED

Exiting WebLogic Scripting Tool.
```

For more information about using WLST commands, see the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

30.3.4.2 Example: Using a Java Program to Scale Out to Servers

To use a Java program to scale out to servers:

1. Open a Java editor and compile a Java program similar to the following example:

```
package com.oracle.bi.example;

import java.util.Hashtable;

import javax.management.Attribute;
import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;
import javax.naming.Context;

/**
 * Example class showing how to connect to the Oracle BIEE AdminMBeans JMX
Interface,
 * and do some scale-out operations.
 * <br/>
 * This example is intentionally succinct for clarity, so for example,
 * it does not do any exception handling.
 * Any production code should include robust error handling.
 * <br/>
 * Note that you need <code>wljmxclient.jar</code> and
<code>wlclient.jar</code> on the classpath.
 */
public class MBeanAPIExample4ScaleOut {
       private String host;
       private int port;
       private String username;
       private String password;
       private String oracleInstanceName;
       private int newNumBIServers;

      private MBeanServerConnection mbs = null;
      private ObjectName biDomainMBeanName;
      private ObjectName biInstanceMBeanName;
      private ObjectName[] oracleInstanceMBeanNames;
      private MBeanAPIExample4ScaleOut(String[] args) {
             if (args.length < 6) {
               System.out.println("Usage: MBeanAPIExample1Connect host port
username password oracleInstanceName NumBIServers");
               throw new IllegalArgumentException("Not enough arguments");
                   }
                   host = args[0];
                   port = Integer.parseInt(args[1]);
                   username = args[2];
                   password = args[3];
                   oracleInstanceName = args[4];
                   newNumBIServers = Integer.parseInt(args[5]);
       }
     private void connect() throws Exception {
              String jmxUrl = "service:jmx:t3://" + host + ":" + port +
"/jndi/weblogic.management.mbeanservers.domainruntime";
              System.out.println("Connecting using URL: " + jmxUrl + " ...");
              Hashtable<String,String> h = new Hashtable<String,String>();
              h.put(Context.SECURITY_PRINCIPAL,   username);
              h.put(Context.SECURITY_CREDENTIALS, password);
              h.put(JMXConnectorFactory.PROTOCOL_PROVIDER_PACKAGES,
"weblogic.management.remote");
              JMXConnector jmxConnector = JMXConnectorFactory.connect(new
JMXServiceURL(jmxUrl), h);
              mbs = jmxConnector.getMBeanServerConnection();
              System.out.println("Connected OK");

              biDomainMBeanName = new
ObjectName("oracle.biee.admin:type=BIDomain,group=Service");
              ObjectName[] biInstanceMBeanNames =
(ObjectName[])mbs.getAttribute(biDomainMBeanName, "BIInstances");
              biInstanceMBeanName = biInstanceMBeanNames[0]; // there should
only be one
              System.out.println("Found BI Instance MBean: " +
biInstanceMBeanName);
oracleInstanceMBeanNames = (ObjectName[])mbs.getAttribute(biDomainMBeanName,
"OracleInstances");
for (ObjectName oracleInstanceMBeanName : oracleInstanceMBeanNames) {
      System.out.println("Found Oracle Instance MBean: " +
oracleInstanceMBeanName);
  }
         }
private void displayStatus() throws Exception {
   String status = (String)mbs.getAttribute(biInstanceMBeanName,
"ServiceStatus");
   System.out.println("Status is: " + status);
}
private void lock() throws Exception {
   System.out.println("Locking domain ...");
   Object[] args = new Object[]{};
   String[] sig  = new String[]{};
   mbs.invoke(biDomainMBeanName, "lock", args, sig);
   System.out.println("Locked domain OK");
}
private void commit() throws Exception {
   System.out.println("Committing changes ...");
   Object[] args = new Object[]{};
   String[] sig  = new String[]{};
   mbs.invoke(biDomainMBeanName, "commit", args, sig);
   System.out.println("Committed changes OK");
}

private ObjectName findMatchingOracleInstanceMBean() throws Exception {
   for (ObjectName oracleInstanceMBeanName : oracleInstanceMBeanNames) {
     String oracleInstanceId =
(String)mbs.getAttribute(oracleInstanceMBeanName, "Id");
```

```
     if (oracleInstanceId.equals(oracleInstanceName)) {
         return oracleInstanceMBeanName;
     }
}
throw new RuntimeException("Cannot find oracle instance: " +
oracleInstanceName);
}

private void scaleOut(ObjectName oracleInstanceMBeanName) throws Exception {
   ObjectName[] biInstanceDeploymentMBeanNames =
(ObjectName[])mbs.getAttribute(oracleInstanceMBeanName,
"BIInstanceDeployments");
   ObjectName biInstanceDeploymentMBeanName =
biInstanceDeploymentMBeanNames[0]; // there should only be one
   System.out.println("Found BI Instance Deployment MBean: " +
biInstanceDeploymentMBeanName);

   int numBIServers = (Integer)mbs.getAttribute(biInstanceDeploymentMBeanName,
"NumBIServers");
   System.out.println("Found Num BI Servers: " + numBIServers);
   System.out.println("Setting Num BI Servers to: " + newNumBIServers);
   mbs.setAttribute(biInstanceDeploymentMBeanName, new
Attribute("NumBIServers", newNumBIServers));
 }

private void stopAll() throws Exception {
   System.out.println("Stopping instance ...");
   Object[] args = new Object[]{};
   String[] sig  = new String[]{};
   mbs.invoke(biInstanceMBeanName, "stop", args, sig);
   System.out.println("Stopped instance OK");
}

private void startAll() throws Exception {
   System.out.println("Starting instance ...");
   Object[] args = new Object[]{};
   String[] sig  = new String[]{};
   mbs.invoke(biInstanceMBeanName, "start", args, sig);
   System.out.println("Started instance OK");
}

      public static void main(String[] args) throws Exception {
                 MBeanAPIExample4ScaleOut example = new
MBeanAPIExample4ScaleOut(args);
                 example.connect();
                 example.displayStatus();
                 ObjectName oiMBeanName =
example.findMatchingOracleInstanceMBean();
                 example.lock();
                 example.scaleOut(oiMBeanName);
                 example.commit();
                 example.stopAll();
                 example.startAll();
                 example.displayStatus();
      }
}
```

2. Navigate to the \<wlserver>\server\lib folder and run the Java program, providing the class name and any arguments required.

For example:

java <*JavaClassName*> <*host*> <*port*> <*username*> <*password*> <oracleInstanceName> <newNumBIServers>

3. Navigate to the \<wlserver>\server\lib folder and run the Java program, providing the class path, class name, and any arguments required.

   java -classpath <*JavaClassName*> <*host*> <*port*> <*username*> <*password*> <*oracleInstanceName*> <*newNumBIServers*>

   For example:

   java -classpath <*MW_HOME*>wls_10.3/server/lib/wjmxclient.jar com.oracle.bi.example.MBeanAPIExample4ScaleOut localhost 7001 user1 password1 instance1 3

   The command window (and log file) displays prompts and messages, for example:

```
Connecting using URL:
service:jmx:t3://localhost:7001/jndi/weblogic.management.mbeanservers.domainrun
time ...
Connected OK
Found BI Instance MBean:
oracle.biee.admin:type=BIDomain.BIInstance,biInstance=coreapplication,group=Ser
vice
Found Oracle Instance MBean:
oracle.biee.admin:oracleInstance=instance1,type=BIDomain.OracleInstance,group=S
ervice
Status is: PARTIALLY_STARTED
Locking domain ...
Locked domain OK
Found BI Instance Deployment MBean:
oracle.biee.admin:oracleInstance=instance1,type=BIDomain.OracleInstance.BIInsta
nceDeployment,biInstance=coreapplication,group=Service
Found Num BI Servers: 1
Setting Num BI Servers to: 3
Committing changes ...
Committed changes OK
Stopping instance ...
Stopped instance OK
Starting instance ...
Started instance OK
Status is: FULLY_STARTED
```

30.4 Descriptions of JMX Admin MBeans and Methods for Oracle BI Systems Management API

This section lists the JMX Admin MBeans of the Oracle BI Systems Management API that you use to perform system administration tasks (using WLST scripts and Java programs). Your code must reference Admin MBeans and their attributes and methods. This section lists the Admin MBeans. The JavaDoc pages describe the attributes and methods that are available for each Admin MBean.

For example, display the JavaDoc files in a Web browser and click the Admin MBean EmailConfigurationMBean. You see all related attributes and methods, and you can then use the correct attribute and method names in WLST scripts or Java programs.

See the JavaDoc file that is located in the following directory:

ORACLE_HOME/doc/javadoc/bifoundation/jmxapi/index.html

Table 30–1 lists the JMX Admin MBeans for the Oracle BI Systems Management API.

Table 30–1 Oracle BI Systems Management API JMX Admin MBeans

Admin MBean	Description
AvailabilityConfigurationMBean	Provides an MBean interface for configuring several options that affect process availability for Oracle Business Intelligence.
BIComponentMBean	Defines an MBean interface for a component of a BI domain.
BIDomainMBean	Defines an MBean interface for managing a BI domain.
BIInstanceDeploymentMBean	Defines an MBean interface to describe the deployment (provisioning) of system components to an Oracle instance.
BIInstanceMBean	Defines an MBean interface for managing system components.
BILogConfigurationMBean	Defines an MBean interface for managing Oracle Business Intelligence log file settings and levels.
BISizeAgeLogConfigurationMBean	Defines an MBean interface for managing query log file settings.
EmailConfigurationMBean	Defines an MBean interface for managing the email configuration of system components.
MarketingConfigurationMBean	Defines an MBean interface for managing the marketing configuration of system components.
OracleInstanceMBean	Defines an MBean interface for managing an Oracle instance.
PerformanceConfigurationMBean	Defines an MBean interface for managing the performance configuration of system components.
PresentationConfigurationMBean	Defines an MBean interface for managing the presentation configuration of system components.
PresentationServerConfigurationMBean	Defines an MBean interface for managing the configuration of the Presentation Server.
ScaleOutSupportMBean	Provides operations to support scaling out a domain.
SchedulerDatabaseMBean	Defines an MBean interface for managing Scheduler database connection settings.
SecurityConfigurationMBean	Defines an MBean interface for managing the centralized security configuration of system components.
ServerConfigurationMBean	Defines an MBean interface for managing settings related to the Oracle BI Server.

You can also use the MBean Browser to access Admin MBeans. For information, see Section 2.4, "Managing Oracle Business Intelligence JMX MBeans Using the Fusion Middleware Control MBean Browser."

Part IX

Managing the Life Cycle

This part describes life cycle management tasks for Oracle Business Intelligence. Life cycle management consists of installing, upgrading, patching, moving to a new environment, moving to a production environment, and backing up and recovering systems.

For information on installation, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

For information on upgrade, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*.

This part contains the following chapters on life cycle management tasks:

- Chapter 31, "Patching Oracle Business Intelligence Systems"
- Chapter 32, "Moving to a New Environment"
- Chapter 33, "Moving From a Test to a Production Environment"
- Chapter 34, "Backup and Recovery of Oracle Business Intelligence Systems"

31

Patching Oracle Business Intelligence Systems

Patching involves copying a small collection of files over an existing installation. A patch is normally associated with a particular version of an Oracle product and involves updating from one minor version of the product to a newer minor version of the same product (for example, from version 11.1.1.2.0 to version 11.1.1.3.0). A patch set is a single patch that contains a collection of patches that are designed to be applied at the same time.

For more information, see the *Oracle Fusion Middleware Patching Guide*.

This chapter contains the following topics:

- Section 31.1, "What is Patched for the Oracle Business Intelligence Platform?"
- Section 31.2, "Patching the Platform"
- Section 31.3, "Rolling Back a Platform Patch"
- Section 31.4, "Determining Current Patch Levels"

31.1 What is Patched for the Oracle Business Intelligence Platform?

Oracle Business Intelligence platform patching applies patches for binary files with extensions such as DLL, JAR, and EXE.

Oracle Business Intelligence platform patching does not patch the following:

- Configuration Files

  If configuration updates are required as part of a patch, then these are detailed in the accompanying README.txt file, and you must manually apply them. No automated mechanism is available for merging customer configuration and patched configuration files.

- Schema-based Metadata

  Non-design-time metadata that is stored in database schemas (including schemas for the Scheduler, usage statistics, event polling, repository files, and the Oracle BI Presentation Catalog) is not patched.

  Other platform metadata (such as repository files and Oracle BI Presentation Catalog files) that are delivered in the context of an application are patched, but as part of an applications patch and not as part of a platform patch.

31.2 Patching the Platform

Typically you apply a patch that contains one or more bug fixes to an existing production Oracle BI EE system that is distributed across one or more computers. Bug fixes might affect the system components and Java components that are deployed inside the Oracle WebLogic Server. The patch might include new server executables and updated and new Java class files.

You use the Oracle OPatch utility to apply (and to roll back) Oracle BI EE platform patches. You download patches from Oracle Support, as described in Section 31.2.2, "Applying a Patch to the Platform."

For more information about patching in Oracle Fusion Middleware, see *Oracle Fusion Middleware Patching Guide*.

This section contains the following topics:

- Section 31.2.1, "What Conditions Apply When Patching the Platform?"
- Section 31.2.2, "Applying a Patch to the Platform"
- Section 31.2.3, "What Happens if a Patching Conflict Occurs?"

31.2.1 What Conditions Apply When Patching the Platform?

The following conditions apply when patching the Oracle BI EE platform:

- Oracle BI EE 11*g* Release 1 (11.1.1) must be installed, and previous versions must have been upgraded to this 11*g* release.
- Oracle Business Intelligence must be stopped before the patch is applied and restarted after all the components are successfully patched.
- If a patch affects existing Java components or includes new Java components, then you must re-deploy the components after applying the patch.

31.2.2 Applying a Patch to the Platform

You must apply a patch to all the related Oracle homes for a given Oracle Business Intelligence system, distributed across multiple computers. Oracle BI EE components must be patched to the same version. You must check the patch version of each Oracle home, one at a time. For information, see Section 31.4, "Determining Current Patch Levels."

Complete the tasks that are described in the following sections to apply a patch:

1. Section 31.2.2.1, "Step 1: Download the Patch"
2. Section 31.2.2.2, "Step 2: Stop System Component Processes for Oracle BI EE"
3. Section 31.2.2.3, "Step 3: Apply the Patch to Each Oracle Home"
4. Section 31.2.2.4, "Step 4: Start System Component Processes for Oracle BI EE"
5. Section 31.2.2.5, "Step 5: Redeploy Patched J2EE Applications"

31.2.2.1 Step 1: Download the Patch

To download patch:

1. Locate the patch (OPatch package) on the Oracle Support Web site at the following URL:

   `http://support.oracle.com`

The patch ID corresponds to a bug or service request number, which is provided by Oracle Support.

2. Download the patch to your local network.

For more information, see *Oracle Fusion Middleware Patching Guide*.

31.2.2.2 Step 2: Stop System Component Processes for Oracle BI EE

For details, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

31.2.2.3 Step 3: Apply the Patch to Each Oracle Home

To apply the patch to each Oracle Home:

1. Display a command window and navigate to the location of the OPatch executable.

   The OPatch utility is located in the *ORACLE_HOME*/*ORACLE_INSTANCE*/OPatch directory (on UNIX operating systems) or *ORACLE_HOME*\ORACLE_INSTANCE\\OPatch directory (on Windows operating systems).

   For example, the directory on Windows might be:

   D:\OBI11g\Oracle_BI1\OPatch

2. Run the OPatch utility using the following command syntax:

   ```
   opatch apply -id <patch_id> -oh <oracle_home>
   ```

   For example, you might use the following command:

   ```
   opatch apply -id 123456 -oh D:\OBI11g
   ```

   For complete information about OPatch command options, see *Oracle Fusion Middleware Patching Guide*.

   OPatch performs the following tasks:

   - Displays the patch number, creation date, bugs fixed, and details of what is updated when the patch is applied.
   - Makes a backup copy of the binary files that are being replaced.
   - Copies new binary files to the correct locations in the Oracle home.
   - Reports that the patch has been successfully applied to the Oracle home.

3. To run the OPatch utility for other Oracle homes, repeat the previous steps.

For more information about applying patches, OPatch requirements, prerequisite checks, and the options for running the utility, see:

- *Oracle Fusion Middleware Patching Guide*
- The README.txt file that you downloaded with the patch update

31.2.2.4 Step 4: Start System Component Processes for Oracle BI EE

For details, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

31.2.2.5 Step 5: Redeploy Patched J2EE Applications

If the patch includes any J2EE applications, then you must redeploy them to the Oracle WebLogic Server instances that they were running on before the patch was applied.

To redeploy patched J2EE applications:

1. Display the Oracle WebLogic Server Administration Console.

   For information, see Section 2.3, "Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console."

2. In the Change Center of the Administration Console, click **Lock & Edit** to enable you to redeploy J2EE applications.
3. In the left pane of the Administration Console, select **Deployments**.
4. In the right pane, select the boxes next to the patched J2EE applications to redeploy.

   This information is included with the patch itself. OPatch lists the JAR files that are affected. The README.txt file details the post-patch steps that are required, which include this manual redeployment step.

5. Click **Update**.
6. In the Update Application Assistant, click **Finish**.
7. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

   Not all changes take effect immediately; some require a restart of the system.

   For more information, see the *Oracle WebLogic Server Administration Console Help* system.

On completion of the patching tasks (and following any necessary restarts), the following conditions apply:

- System component servers have been patched and re-started and are fully operational.
- Backup versions of changed items have been stored in the Oracle home directory and are ready for potential rollback.
- Patch history, archives, and logs have been updated.

31.2.3 What Happens if a Patching Conflict Occurs?

If a patching conflict occurs, then the process stops and you should contact Oracle Support Services.

31.3 Rolling Back a Platform Patch

OPatch maintains metadata for each patch that is applied to each Oracle home and keeps copies of what it replaces during a patch. Therefore, you can roll back a complete patch.

> **Note:** To confirm that an Oracle BI EE platform patch is no longer applied after a rollback, you must establish the patch levels before applying the rollback, then repeat the task after rollback. For information, see Section 31.4, "Determining Current Patch Levels."

Complete the tasks that are described in the following sections to apply a patch:

- Section 31.3.1, "Step 1: Stop System Component Processes for Oracle BI EE"
- Section 31.3.2, "Step 2: Invoke OPatch with "-rollback" Option"

- Section 31.3.3, "Step 3: Start System Component Processes for Oracle BI EE"
- Section 31.3.4, "Step 4: Redeploy J2EE Applications"

31.3.1 Step 1: Stop System Component Processes for Oracle BI EE

For details, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

31.3.2 Step 2: Invoke OPatch with "-rollback" Option

You must invoke OPatch with the "-rollback" option for each Oracle home while specifying the ID of the patch to be undone.

To invoke OPatch with the "-rollback" option:

1. Display a command window and navigate to the location of the OPatch executable.

   The OPatch utility is located in the *ORACLE_HOME*/*ORACLE_INSTANCE*/OPatch directory (on UNIX operating systems) or *ORACLE_HOME*\ORACLE_INSTANCE\\OPatch directory (on Windows operating systems).

   For example, the directory on Windows might be:

   D:\OBI11g\Oracle_BI1\OPatch

2. Run the OPatch utility with the -rollback option using the following command syntax:

   ```
   opatch rollback -id <patch_id> -oh <oracle_home>
   ```

   For example, you might use the following command:

   ```
   opatch rollback -id 123456 -oh D:\OBI11g\Oracle_BI1\OPatch
   ```

For more information about rolling back patches, OPatch requirements, and the options for running the utility, see:

- *Oracle Fusion Middleware Patching Guide*
- The README.txt file that was downloaded with the patch update

31.3.3 Step 3: Start System Component Processes for Oracle BI EE

For details, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

31.3.4 Step 4: Redeploy J2EE Applications

If the patch included any J2EE applications, then you must redeploy them to the Oracle WebLogic Server instances that they were running on before the patch was rolled back.

To redeploy patched J2EE applications:

1. Display the Oracle WebLogic Server Administration Console.

   For information, see Section 2.3, "Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console."

2. In the Change Center of the Administration Console, click **Lock & Edit** to enable you to redeploy J2EE applications.
3. In the left pane of the Administration Console, select **Deployments**.
4. In the right pane, select the boxes next to the J2EE applications to redeploy.

5. Click **Update**.
6. In the Update Application Assistant, click **Finish**.
7. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

   Not all changes take effect immediately; some require a system restart.

   For more information, see the *Oracle WebLogic Server Administration Console Help* system.

Following completion of the rollback, the binary and JAR files that were updated by the application of the previous patch are now returned to their original state.

31.4 Determining Current Patch Levels

Each Oracle home must be patched to the same version as OPatch to ensure that Oracle BI EE functions properly. Use the OPatch lsinventory utility to determine the current patch version for any given Oracle home in the system. You can also use the utility to retrieve a full list of patches, with their corresponding IDs, for a given Oracle home.

To determine the current patch levels:

1. Display a command window and navigate to the location of the OPatch executable:

   ORACLE_HOME\ORACLE_INSTANCE\\OPatch

   For example:

   D:\\OBI11g\\Oracle_BI1\\OPatch
2. Run the lsinventory utility using the following command syntax:

   ```
   <Path_to_OPatch>/opatch lsinventory [-all] [-detail] [-patch]
   [-oh (Oracle home location)]
   ```

   For example:

   opatch lsinventory -patch -detail

   For information about the lsinventory options, see the user guides in the *ORACLE_HOME*\ORACLE_INSTANCE\\OPatch\\docs directory.
3. To run the lsinventory utility against other Oracle homes, repeat the previous steps for each Oracle home.

For more information, see *Oracle Fusion Middleware Patching Guide*.

32 Moving to a New Environment

You can move (or migrate) Oracle Business Intelligence to a new environment by re-creating an existing Oracle BI system in a different location to the one in which it was originally installed. The objective is to re-create an identical deployment on different hardware.

You might want to move to a new environment for the following reasons:

- To move the system as a whole onto more powerful hardware.
- To move to a different operating system.
- To move into a different physical location.

For information, see "Moving Oracle Business Intelligence Components to a Production Environment" in *Oracle Fusion Middleware Administrator's Guide*.

Moving can also be completed as part of a upgrade. Upgrade is not covered in this guide. For complete information, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*.

33 Moving From a Test to a Production Environment

Moving from a test environment to a production environment involves developing and testing applications in a test environment, and then rolling out the test applications (and optionally test data), in the production environment. This can also include moving from a single- to a multiple-computer environment.

For details about moving Oracle Business Intelligence from a test environment to a production environment, see *Oracle Fusion Middleware Administrator's Guide*.

See also "Managing the Repository Lifecycle in a Multiuser Development Environment" in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about moving repositories from test to production environments.

34

Backup and Recovery of Oracle Business Intelligence Systems

Backup and recovery refers to the various strategies and procedures involved in guarding against hardware failures and data loss and in reconstructing data should loss occur.

Backup and recovery for Oracle Business Intelligence is fully described in *Oracle Fusion Middleware Administrator's Guide*.

Part X

Reference Information

This part provides reference information for managing Oracle Business Intelligence. It contains the following appendixes:

- Appendix A, "NQSConfig.INI File Configuration Settings"
- Appendix B, "Exchanging Metadata with Databases to Enhance Query Performance"
- Appendix C, "Managing Security for Dashboards and Analyses"
- Appendix D, "Setting Up the SA System Subject Area"
- Appendix E, "Advanced Configuration Reference"
- Appendix F, "Propagating Labels and Tooltips from Oracle ADF Data Sources"
- Appendix G, "Mapping User Interface Labels with Configuration File Elements"
- Appendix H, "Manually Updating Fusion Middleware Control Settings"

A

NQSConfig.INI File Configuration Settings

The Oracle BI Server software uses an initialization file called NQSConfig.INI to set parameters upon startup. This initialization file includes parameters to customize behavior based on the requirements of each individual installation. This appendix lists the NQSConfig.INI file parameters and gives a brief description and any required syntax for each parameter. The parameters are generally listed in the order they appear in the configuration file.

> **Note:** The examples in this appendix assume that you are editing a Windows version of NQSConfig.INI. If you are editing this file on a UNIX system, then ensure that you use UNIX-appropriate file system paths and conventions.

This appendix contains the following topics:

- Section A.1, "About Parameters in the NQSConfig.INI File"
- Section A.2, "Repository Section Parameters"
- Section A.3, "Query Result Cache Section Parameters"
- Section A.4, "General Section Parameters"
- Section A.5, "Security Section Parameters"
- Section A.6, "Server Section Parameters"
- Section A.7, "Dynamic Library Section Parameters"
- Section A.8, "Usage Tracking Section Parameters"
- Section A.9, "Query Optimization Flags Section Parameters"
- Section A.10, "MDX Member Name Cache Section Parameters"
- Section A.11, "Aggregate Persistence Section Parameters"
- Section A.12, "JavaHost Section Parameters"

A.1 About Parameters in the NQSConfig.INI File

Each instance of the Oracle BI Server has its own NQSConfig.INI file. When you update NQSConfig.INI parameters for a clustered deployment, ensure that you make the change in the NQSConfig.INI file for each instance of the Oracle BI Server.

Some parameters in NQSConfig.INI are centrally managed by Fusion Middleware Control and cannot be updated manually in NQSConfig.INI. Instead, use Fusion Middleware Control to change these parameters. Parameters that are centrally managed by Fusion Middleware Control are marked as such in this appendix, and are also identified by comments in the NQSConfig.INI file.

See Appendix G, "Mapping User Interface Labels with Configuration File Elements" for additional information.

> **Note:** If you attempt to manually update NQSConfig.INI parameters that are centrally managed by Fusion Middleware Control, then the manually updated values are ignored, because the value set in Fusion Middleware Control overrides the value in the file. If you must manually update these settings, then you must disable all configuration through Fusion Middleware Control (not recommended).

Note the following rules and guidelines for NQSConfig.INI file entries:

- The Oracle BI Server reads the NQSConfig.INI file each time it is started.
- Each parameter entry in NQSConfig.INI must be within the section to which the parameter belongs (Repository, Cache, General, and so on).
- Each entry must be terminated with semicolon (;).
- You can add comments anywhere in the NQSConfig.INI file. Comments must begin with either of the following:

  ```
  #
  ```

  ```
  //
  ```

  Any text following these comment characters up to the end of the line is ignored when the Oracle BI Server reads the file.
- Any syntax errors prevent the Oracle BI Server from starting. The errors are logged to the nqserver.log file, which is located in:

  ```
  ORACLE_INSTANCE/diagnostics/logs/OracleBIServerComponent/coreapplication_obisn
  ```

  There might also be a summary message in the system log that relates to the error.

  If you get an error, then correct the problem and start the Oracle BI Server again. Repeat this process until the server starts with no errors.

A.1.1 How to Update Parameters in NQSConfig.INI

The following procedure explains how to update parameters in NQSConfig.INI.

To update parameters in NQSConfig.INI:

1. Open the NQSConfig.INI file in a text editor. You can find this file at:

   ```
   ORACLE_INSTANCE/config/OracleBIServerComponent/coreapplication_obisn
   ```

   Make a backup copy of the file before editing.
2. Locate and update the parameter you want to change.
3. Save and close the file.
4. Restart the Oracle BI Server. For more information, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

5. If you have multiple Oracle BI Server instances, then repeat these steps in each NQSConfig.INI file for all Oracle BI Server instances.

A.2 Repository Section Parameters

> **Note:** The default repository is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).
>
> The Repository tab of the Deployment page in Fusion Middleware Control controls the default repository. The logical name of the default repository is always "Star." See Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location" for more information.
>
> Note that you can manually add additional repository entries in NQSConfig.INI without having these entries overridden by Fusion Middleware Control. However, hosting multiple repositories on a single Oracle BI Server is not recommended for production systems.

The Repository section contains one entry for every repository that is loaded when the server starts.

Syntax: `logical_name = repository_name.rpd;`

Optional syntax: `logical_name = repository_name.rpd, DEFAULT;`

where:

- *`logical_name`*: A logical name for the repository. Client tools use this name to configure the ODBC data sources that connect to the repository. To use a reserved keyword for the name, such as OCI7 or OCI8, enclose it in single quotes.
- *`repository_name.rpd`*: The file name of the repository. The file name must have the .rpd file extension, and the file must reside in the repository subdirectory.

  The demonstration repository file, SampleApp.rpd, is installed with the Simple installation type.

When `DEFAULT` is specified for a repository, connections that do not specify a logical repository name in the DSN connect to the default repository.

Example: `Star = SampleApp.rpd, DEFAULT;`

A.3 Query Result Cache Section Parameters

The parameters in the Query Result Cache Section provide configuration information for Oracle BI Server caching. The query cache is enabled by default. After deciding on a strategy for flushing outdated entries, you should configure the cache storage parameters in Fusion Middleware Control and in the NQSConfig.INI file.

Note that query caching is primarily a run-time performance improvement capability. As the system is used over a period of time, performance tends to improve due to cache hits on previously executed queries. The most effective and pervasive way to optimize query performance is to use the Aggregate Persistence Wizard and aggregate navigation.

This section describes only the parameters that control query caching. For information about how to use caching in Oracle Business Intelligence, including information about how to use agents to seed the Oracle BI Server cache, see Chapter 7, "Managing Performance Tuning and Query Caching."

A.3.1 ENABLE

Note: The `ENABLE` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

The **Cache enabled** option on the Performance tab of the Capacity Management page in Fusion Middleware Control corresponds to the `ENABLE` parameter. See Section 7.5.1, "Using Fusion Middleware Control to Enable and Disable Query Caching" for more information.

Specifies whether the cache system is enabled. When set to `NO`, caching is disabled. When set to `YES`, caching is enabled. The query cache is enabled by default.

Example: `ENABLE = YES;`

A.3.2 DATA_STORAGE_PATHS

Specifies one or more paths for where the cached query results data is stored and are accessed when a cache hit occurs and the maximum capacity in bytes, kilobytes, megabytes, or gigabytes. The maximum capacity for each path is 4 GB. For optimal performance, the paths specified should be on high performance storage systems.

Each path listed must be an existing, writable path name, with double quotes (") surrounding the path name. Specify mapped directories only. UNC path names and network mapped drives are allowed only if the service runs under a qualified user account.

You can specify either fully qualified paths, or relative paths. When you specify a path that does not start with "/" (on UNIX) or "<drive>:" (on Windows), the Oracle BI Server assumes that the path is relative to the local writable directory. For example, if you specify the path "cache," then at run time, the Oracle BI Server uses the following:

```
ORACLE_INSTANCE/bifoundation/OracleBIServerComponent/coreapplication_obisn/cache
```

Note: Multiple Oracle BI Servers across a cluster do not share cached data. Because of this, the `DATA_STORAGE_PATHS` entry must be unique for each clustered server. To ensure this, enter a relative path so that cache is stored in the local writable directory for each Oracle BI Server, or enter different fully qualified paths for each server.

Specify multiple directories with a comma-delimited list. When you specify multiple directories, they should reside on different physical drives. (If you have multiple cache directory paths that all resolve to the same physical disk, then both available and used space might be double-counted.)

Syntax: `DATA_STORAGE_PATHS = "path_1" sz[, "path_2" sz{, "path_n" sz}];`

Example: `DATA_STORAGE_PATHS = "cache" 256 MB;`

> **Note:** Specifying multiple directories for each drive does not improve performance, because file input and output (I/O) takes place through the same I/O controller. In general, specify only one directory for each disk drive. Specifying multiple directories on different drives might improve the overall I/O throughput of the Oracle BI Server internally by distributing I/O across multiple devices.

The disk space requirement for the cached data depends on the number of queries that produce cached entries, and the size of the result sets for those queries. The query result set size is calculated as row size (or the sum of the maximum lengths of all columns in the result set) times the result set cardinality (that is, the number of rows in the result set). The expected maximum should be the guideline for the space needed.

This calculation gives the high-end estimate, not the average size of all records in the cached result set. Therefore, if the size of a result set is dominated by variable length character strings, and if the length of those strings are distributed normally, you would expect the average record size to be about half of the maximum record size.

> **Note:** It is a best practice to use a value that is less than 4 GB. Otherwise, the value might exceed the maximum allowable value for an unsigned 32-bit integer, because values over 4 GB cannot be processed on 32-bit systems. It is also a best practice to use values less than 4 GB on 64-bit systems.
>
> Create multiple paths if you have values in excess of 4 GB.

A.3.3 MAX_ROWS_PER_CACHE_ENTRY

Specifies the maximum number of rows in a query result set to qualify for storage in the query cache. Limiting the number of rows is a useful way to avoid consuming the cache space with runaway queries that return large numbers of rows. If the number of rows a query returns is greater than the value specified in the `MAX_ROWS_PER_CACHE_ENTRY` parameter, then the query is not cached.

When set to 0, there is no limit to the number of rows per cache entry.

Example: `MAX_ROWS_PER_CACHE_ENTRY = 100000;`

A.3.4 MAX_CACHE_ENTRY_SIZE

> **Note:** The `MAX_CACHE_ENTRY_SIZE` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).
>
> The **Maximum cache entry size** option on the Performance tab of the Capacity Management page in Fusion Middleware Control corresponds to the `MAX_CACHE_ENTRY_SIZE` parameter. See Section 7.5.2, "Using Fusion Middleware Control to Set Query Cache Parameters" for more information.

Specifies the maximum size for a cache entry. Potential entries that exceed this size are not cached. The default size is 20 MB.

Specify GB for gigabytes, KB for kilobytes, MB for megabytes, and no units for bytes.

Example: `MAX_CACHE_ENTRY_SIZE = 20 MB;`

A.3.5 MAX_CACHE_ENTRIES

Note: The `MAX_CACHE_ENTRIES` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

The **Maximum cache entries** option on the Performance tab of the Capacity Management page in Fusion Middleware Control corresponds to the `MAX_CACHE_ENTRIES` parameter. See Section 7.5.2, "Using Fusion Middleware Control to Set Query Cache Parameters" for more information.

Specifies the maximum number of cache entries allowed in the query cache to help manage cache storage. The actual limit of cache entries might vary slightly depending on the number of concurrent queries. The default value is 1000.

Example: `MAX_CACHE_ENTRIES = 1000;`

A.3.6 POPULATE_AGGREGATE_ROLLUP_HITS

Specifies whether to aggregate data from an earlier cached query result set and create a new entry in the query cache for rollup cache hits. The default value is `NO`.

Typically, if a query gets a cache hit from a previously executed query, then the new query is not added to the cache. A user might have a cached result set that contains information at a particular level of detail (for example, sales revenue by ZIP code). A second query might ask for this same information, but at a higher level of detail (for example, sales revenue by state). The `POPULATE_AGGREGATE_ROLLUP_HITS` parameter overrides this default when the cache hit occurs by rolling up an aggregate from a previously executed query (in this example, by aggregating data from the first result set stored in the cache). That is, Oracle Business Intelligence sales revenue for all ZIP codes in a particular state can be added to obtain the sales revenue by state. This is referred to as a rollup cache hit.

Normally, a new cache entry is not created for queries that result in cache hits. You can override this behavior specifically for cache rollup hits by setting `POPULATE_AGGREGATE_ROLLUP_HITS` to `YES`. Nonrollup cache hits are not affected by this parameter. If a query result is satisfied by the cache—that is, the query gets a cache hit—then this query is not added to the cache. When this parameter is set to `YES`, then when a query gets an aggregate rollup hit, then the result is put into the cache. Setting this parameter to `YES` might result in better performance, but results in more entries being added to the cache.

Example: `POPULATE_AGGREGATE_ROLLUP_HITS = NO;`

A.3.7 USE_ADVANCED_HIT_DETECTION

When caching is enabled, each query is evaluated to determine whether it qualifies for a cache hit. A cache hit means that the server was able to use cache to answer the query and did not go to the database at all. The Oracle BI Server can use query cache to answer queries at the same or later level of aggregation.

The parameter `USE_ADVANCED_HIT_DETECTION` enables an expanded search of the cache for hits. The expanded search has a performance impact, which is not easily quantified because of variable customer requirements. Customers that rely heavily on query caching and are experiencing misses might want to test the trade-off between better query matching and overall performance for high user loads. See also the parameter "MAX_SUBEXPR_SEARCH_DEPTH" for related information.

A.3.7.1 Reasons Why a Query is Not Added to the Cache

Customers who rely on query result caching in the Oracle BI Server to meet their performance KPIs can use caching parameters to help determine why a cache hit did not occur. Logging facilities can help diagnose common reasons for getting a cache miss, where the logical SQL query that was supposed to seed the cache did not get inserted into the cache. The following describes some situations when this might occur.

- Non-cacheable SQL element. If a SQL request contains `CURRENT_TIMESTAMP`, `CURRENT_TIME`, `RAND`, `POPULATE`, or a parameter marker, then it is not added to the cache.
- Non-cacheable table. Physical tables in the Oracle BI Server repository can be marked 'non-cacheable.' If a query references any non-cacheable table, then the query results are not added to the cache.
- Cache hit. In general, if the query gets a cache hit on a previously cached query, then the results of the current query are not added to the cache.

  The exception is query hits that are aggregate roll-up hits. These are added to the cache if the NQSConfig.INI parameter `POPULATE_AGGREGATE_ROLLUP_HITS` has been set to `YES`.
- Result set is too big.

  This situation occurs when you exceed the size set in `DATA_STORAGE_PATHS`, or if you have rows in excess of the number set in `MAX_ROWS_PER_CACHE_ENTRY`. See Section A.3.2, "DATA_STORAGE_PATHS" and Section A.3.3, "MAX_ROWS_PER_CACHE_ENTRY" for more information.
- Query is canceled. This can happen by explicit cancellation from Oracle BI Presentation Services or the Administration Tool, or implicitly through timeout.
- Oracle BI Server is clustered. Queries that fall into the 'cache seeding' family are propagated throughout the cluster. Other queries continue to be stored locally. Therefore, even though a query might be put into the cache on Oracle BI Server node 1, it might not be on Oracle BI Server node 2.

Level 4 of query logging is the best tool to diagnose whether the Oracle BI Server compiler intended to add the entry into the query result cache. See Section 8.3.1, "Configuring Query Logging" for more information.

A.3.8 MAX_SUBEXPR_SEARCH_DEPTH

Lets you configure how deep the hit detector looks for an inexact match in an expression of a query. The default is `5`.

For example, at level 5, a query on the expression `SIN(COS(TAN(ABS(ROUND(TRUNC(profit))))))` misses on `profit`, which is at level 7. Changing the search depth to 7 opens up `profit` for a potential hit.

A.3.9 DISABLE_SUBREQUEST_CACHING

When set to `YES`, disables caching at the subrequest (subquery) level. The default value is `NO`.

Caching subrequests improves performance and the cache hit ratio, especially for queries that combine real-time and historical data. In some cases, however, subrequest caching might need to be disabled, such as when other methods of query optimization provide better performance.

Example: `DISABLE_SUBREQUEST_CACHING = NO;`

A.3.10 GLOBAL_CACHE_STORAGE_PATH

> **Note:** The `GLOBAL_CACHE_STORAGE_PATH` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).
>
> The **Global cache path** and **Global cache size** options on the Performance tab of the Capacity Management page in Fusion Middleware Control correspond to the `GLOBAL_CACHE_STORAGE_PATH` parameter. See Section 7.5.4, "Using Fusion Middleware Control to Set Global Cache Parameters" for more information.

In a clustered environment, Oracle BI Servers can be configured to access a shared cache that is referred to as the global cache. The global cache resides on a shared file system storage device and stores seeding and purging events and the result sets that are associated with the seeding events.

This parameter specifies the physical location for storing cache entries shared across clustering. This path must point to a network share. All clustering nodes share the same location.

You can specify the size in KB, MB, or GB, or enter a number with no suffix to specify bytes.

Syntax: `GLOBAL_CACHE_STORAGE_PATH = "`*`directory name`*`" SIZE;`

Example: `GLOBAL_CACHE_STORAGE_PATH = "C:\cache" 250 MB;`

A.3.11 MAX_GLOBAL_CACHE_ENTRIES

The maximum number of cache entries stored in the location that is specified by `GLOBAL_CACHE_STORAGE_PATH`.

Example: `MAX_GLOBAL_CACHE_ENTRIES = 1000;`

A.3.12 CACHE_POLL_SECONDS

The interval in seconds that each node polls from the shared location that is specified in `GLOBAL_CACHE_STORAGE_PATH`.

Example: `CACHE_POLL_SECONDS = 300;`

A.3.13 CLUSTER_AWARE_CACHE_LOGGING

Turns on logging for the cluster caching feature. Used only for troubleshooting. The default is `NO`.

Example: `CLUSTER_AWARE_CACHE_LOGGING = NO;`

A.4 General Section Parameters

The General section contains general server default parameters, including localization and internationalization, temporary space and memory allocation, and other default parameters used to determine how data is returned from the Oracle BI Server to a client.

> **Note:** The settings for the parameters `LOCALE`, `SORT_ORDER_LOCALE`, `SORT_TYPE` and `CASE_SENSITIVE_CHARACTER_COMPARISON`, described in the following topics, are interrelated. They help determine how the Oracle BI Server sorts data.

A.4.1 LOCALE

Specifies the locale in which data is returned from the server. This parameter also determines the localized names of days and months.

To successfully run Oracle Business Intelligence, ensure that you configure the appropriate locales on the operating system for the language in which you run the applications. (In some cases, you might need to install additional content on your system to support the locale.) The Oracle BI Server sets the C-runtime locale during the server startup. Some locale- and language-related settings are interrelated and help determine how the Oracle BI Server sorts data. Ensure that the settings for the following parameters work together:

- `LOCALE`
- `SORT_ORDER_LOCALE`
- `SORT_TYPE`
- `CASE_SENSITIVE_CHARACTER_COMPARISON`

Valid platform-independent values for `LOCALE` and `SORT_ORDER_LOCALE` are:

- Arabic
- Chinese
- Chinese-traditional
- Croatian
- Czech
- Danish
- Dutch
- English-USA
- Finnish
- French
- German
- Greek
- Hebrew
- Hungarian

- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese
- Portuguese-Brazilian
- Romanian
- Russian
- Slovak
- Spanish
- Swedish
- Thai
- Turkish

For information about Oracle BI Catalog Manager and language extensions, see Chapter 16, "Localizing Oracle Business Intelligence".

A.4.2 SORT_ORDER_LOCALE

Used to help determine whether the Oracle BI Server can function-ship (push down) an `ORDER BY` clause to a relational database. `ORDER BY` clauses are used in sorting.

Every database that is defined in the Physical layer in the Oracle BI Administration Tool has a features table associated with it. If you want to override the default value in the Features table for a particular type of relational database, then you must do so for all occurrences of it in the Physical layer.

In the Oracle BI Administration Tool, the Features table in the Features tab of the Database dialog specifies the features and functions that the relational database supports. The settings for `SORT_ORDER_LOCALE` in the Features table and in the NQSConfig.INI file should match only if the database and the Oracle BI Server sort data in the same way.

For the relational database and the Oracle BI Server to sort data the same way, they must be in agreement on the parameters that are shown in Table A–1.

Table A–1 Critical SORT_ORDER_LOCALE Parameters

Functional Category	Specific Parameters
Base language	`LOCALE`
Base language	`SORT_ORDER_LOCALE` The default value for `SORT_ORDER_LOCALE` in both the Features table and in the NQSConfig.INI file is `english-usa`. If the Oracle BI Server and the database sort data differently, then the Features table entry `SORT_ORDER_LOCALE` for the database must be set to a different value than `english-usa`. Otherwise, the different data sort methods clash. **Note**: The `LOCALE` and `SORT_ORDER_LOCALE` parameters accept platform-independent names only. See the list provided in Section A.4.1, "LOCALE" for details.

Table A–1 (Cont.) Critical SORT_ORDER_LOCALE Parameters

Functional Category	Specific Parameters
Case	CASE_SENSITIVE_CHARACTER_COMPARISON
Binary versus linguistic comparison	SORT_TYPE

If the SORT_ORDER_LOCALE setting in the actual data source does not match the SORT_ORDER_LOCALE setting in the Features tab of the Database dialog box in the Oracle BI repository, then result sets might not be correct. If the settings do not match, then incorrect answers can result when using multi-database joins, or errors can result when using the Union, Intersect, and Except operators, which all rely on consistent sorting between the back-end data source and the Oracle BI Server.

If the SORT_ORDER_LOCALE setting in NQSConfig.INI does not match the SORT_ORDER_LOCALE setting in the Features tab of the Database dialog box in the Oracle BI repository, then query performance might be negatively impacted. However, this situation does not affect the correctness of the result set.

Example: `SORT_ORDER_LOCALE = "english-usa";`

A.4.2.1 SORT_ORDER_LOCALE on UNIX Operating Systems

The Oracle BI Server sets the C-runtime locale during server startup. A value for the setting is specified using the SORT_ORDER_LOCALE entry in the NQSConfig.INI file. See Chapter 16.3.1, "Setting Locale Parameters on Oracle BI Server" for more information.

A.4.3 SORT_TYPE

Specifies the type of sort to perform. The default value is BINARY. Binary sorts are faster than nonbinary sorts.

Valid values are BINARY and DEFAULT. If you specify DEFAULT, then a nonbinary sort is performed; this yields better sort results for data that contains accented characters.

Example: `SORT_TYPE = "BINARY";`

A.4.4 CASE_SENSITIVE_CHARACTER_COMPARISON

Specifies whether the Oracle BI Server differentiates between uppercase and lowercase characters when performing comparison operations.

Valid values are ON and OFF. When set to OFF, case is ignored. When set to ON, case is considered for comparisons. This parameter is set to ON by default. For binary sorts, case sensitivity for the server and for the relational database should be set the same way.

For information on how this parameter relates to the case setting in Oracle BI Presentation Services, see Section E.1, "Making Advanced Configuration Changes for Oracle BI Presentation Services."

This setting only applies to the internal comparisons of the Oracle BI Server for caching and aggregation. Case sensitivity is a function of database operations and is set at the database level. The CASE_SENSITIVE_CHARACTER_COMPARISON parameter allows the Oracle BI Server to match the functions of the back-end database. The following operators are affected:

- Order By

- Group By
- Distinct
- Join
- comparisons (<, >, =, <=, >=, <>)

For example, consider the following three terms:

- ACME
- DELTA
- acme

An ORDER BY with CASE_SENSITIVE_CHARACTER_COMPARISON set to ON results in rows in the order shown in the preceding example. An ORDER BY with a case-insensitive setting results in ACME and acme appearing next to one another in the list.

If the term is case-sensitive and you perform a duplicate remove (DISTINCT), then the result is three rows. If the term is not case-sensitive, then the DISTINCT result is two rows.

CASE_SENSITIVE_CHARACTER_COMPARISON should be set to correspond with how the back-end database deals with case. For example, if the back-end database is case-insensitive, then the Oracle BI Server should be configured to be case-insensitive. If the Oracle BI Server and the back-end database are not similarly case-sensitive, then some subtle problems can result.

For an example of CASE_SENSITIVE_CHARACTER_COMPARISON applied to aggregation, a case-sensitive database has the following tuples (or rows):

```
Region  Units
 WEST     1
 west     1
 West     1
```

With CASE_SENSITIVE_CHARACTER_COMPARISON set to ON, the data is returned to the client the with the same results shown in the preceding table.

With CASE_SENSITIVE_CHARACTER_COMPARISON set to OFF, the data is again returned to the client the with the same results shown in the preceding table. There is no change because the Oracle BI Server has not done any character comparisons.

However, if SUM_SUPPORTED is set to OFF in the features table, the Oracle BI Server is forced to do a character comparison. The results of the query in this case are as follows:

```
Region  Units
 WEST     3
```

The reason for these results is that the Oracle BI Server has case-sensitive character comparison turned off, so it now treats the three tuples as the same value and aggregates them. In this case WEST = West = west. However, if you filter on the Region column, you would still see the regions WEST, West, and west; CASE_SENSITIVE_CHARACTER_COMPARISON does not affect filtering on a back-end database. The logic shown in the aggregation example applies to caching as well.

Because CASE_SENSITIVE_CHARACTER_COMPARISON is set in the NQSConfig.INI file, the parameter applies to all back-end databases in a repository. Therefore, it should be set to match the case sensitivity of the dominant back-end database of the repository.

Example: `CASE_SENSITIVE_CHARACTER_COMPARISON = ON;`

A.4.5 NULL_VALUES_SORT_FIRST

Specifies if `NULL` values sort before other values (`ON`) or after (`OFF`). `ON` and `OFF` are the only valid values. The value of `NULL_VALUES_SORT_FIRST` should conform to the underlying database. If there are multiple underlying databases that sort `NULL` values differently, then set the value to correspond to the database that is used the most in queries.

Example: `NULL_VALUES_SORT_FIRST = OFF;`

A.4.6 DATE_TIME_DISPLAY_FORMAT

Specifies the format for how date/time stamps are input to and output from the Oracle BI Server. The default value is yyyy/mm/dd hh:mi:ss.

Example: `DATE_TIME_DISPLAY_FORMAT = "yyyy/mm/dd hh:mi:ss";`

A.4.7 DATE_DISPLAY_FORMAT

Specifies the format for how dates are input to and output from the Oracle BI Server. The default value is yyyy/mm/dd.

> **Note:** Specify the year as either 2-digit (yy) or 4-digit (yyyy). Separators can be any character except y, m, or d.

Example: `DATE_DISPLAY_FORMAT = "yyyy/mm/dd";`

A.4.8 TIME_DISPLAY_FORMAT

Specifies the format for how times are input to and output from the Oracle BI Server. The default value is hh:mi:ss.

Example: `TIME_DISPLAY_FORMAT = "hh:mi:ss";`

A.4.9 WORK_DIRECTORY_PATHS

Specifies one or more directories for temporary space.

Each directory listed must be an existing, writable path name, with double quotes (") surrounding the path name. Specify mapped directories only.

You can specify either fully qualified paths, or relative paths. When you specify a path that does not start with "/" (on UNIX) or "<drive>:" (on Windows), the Oracle BI Server assumes that the path is relative to the local writable directory. For example, if you specify the path "temp," then at run time, the Oracle BI Server uses the following:

```
ORACLE_INSTANCE/tmp/OracleBIServerComponent/coreapplication_obisn/temp
```

Specify multiple directories with a comma-delimited list. Valid values are any relative path, or fully qualified path to an existing, writable directory. UNC path names and network mapped drives are allowed only if the service runs under a qualified user account.

For optimum performance, temporary directories should reside on high-performance storage devices. If you specify multiple directories, then they should reside on different physical drives.

Syntax: `WORK_DIRECTORY_PATHS = "path_1" [, "path_2"{, "path_n"}];`

Example 1: `WORK_DIRECTORY_PATHS = "temp" ;`

Example 2: `WORK_DIRECTORY_PATHS = "D:\temp", "F:\temp";`

> **Note:** Specifying multiple directories for each drive does not improve performance because file I/O takes place through the same I/O controller. In general, specify only one directory for each disk drive. Specifying multiple directories on different drives improves the overall I/O throughput of the Oracle BI Server because internally, the processing files are allocated using a round-robin algorithm that balances the I/O load across the given disk drives.

A.4.10 VIRTUAL_TABLE_PAGE_SIZE

Several operations, such as sort, join, union, and database fetch, can require memory resources beyond those available to the Oracle BI Server. To manage this condition, the server uses a virtual table management mechanism that provides a buffering scheme for processing these operations. When the amount of data exceeds the `VIRTUAL_TABLE_PAGE_SIZE`, the remaining data is buffered in a temporary file and placed in the virtual table as processing continues. This mechanism supports dynamic memory sizes and ensures that any row can be obtained dynamically for processing queries.

`VIRTUAL_TABLE_PAGE_SIZE` specifies the size of a memory page for Oracle BI Server internal processing. A higher value reduces I/O but increases memory usage, especially in a multiuser environment.

When `VIRTUAL_TABLE_PAGE_SIZE` is increased, I/O operations are reduced. Complex queries might use 20 to 30 virtual tables, while simple queries might not even require virtual tables. The default size of 128 KB is a reasonable size when one considers that the size for virtual paging in Windows NT is 64 KB. This parameter can be tuned depending on the number of concurrent users and the average query complexity. In general, setting the size higher than 256 KB does not yield a corresponding increase in throughput due to the 64 KB size limit of Windows NT system buffers, as each I/O still goes through the system buffers. 128 KB is also a reasonable value on UNIX systems.

Example: `VIRTUAL_TABLE_PAGE_SIZE = 128 KB;`

A.4.11 USE_LONG_MONTH_NAMES

Specifies whether month names are returned as full names, such as `JANUARY` and `FEBRUARY`, or as three-letter abbreviations, such as `JAN` and `FEB`. Valid values are `YES` and `NO`. Specify `YES` to have month names returned as full names, or `NO` to have months names returned as three-letter abbreviations. The default value is `NO`.

Example: `USE_LONG_MONTH_NAMES = NO;`

A.4.12 USE_LONG_DAY_NAMES

Specifies whether day names are returned as full names, such as `MONDAY` and `TUESDAY`, or as three-letter abbreviations, such as `MON` and `TUE`. Valid values are `YES` and `NO`. Specify `YES` to have day names returned as full names, or `NO` to have day names returned as three-letter abbreviations. The default value is `NO`.

Example: `USE_LONG_DAY_NAMES = NO;`

A.4.13 UPPERCASE_USERNAME_FOR_INITBLOCK

You can use the special syntax `:USER` in initialization blocks to pass through user names. When this parameter is set to `YES`, then user names passed through initialization blocks using `:USER` are changed to all uppercase. Otherwise, case is maintained in the user names.

Example: `UPPERCASE_USERNAME_FOR_INITBLOCK = NO;`

A.5 Security Section Parameters

The security parameters specify default values for the Oracle BI Server security features. For more information about security, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

A.5.1 DEFAULT_PRIVILEGES

Specifies the privileges that users and groups are assigned when they are initially created.

Valid values are `NONE` and `READ`. The default value is `READ`.

Example: `DEFAULT_PRIVILEGES = READ;`

A.5.2 PROJECT_INACCESSIBLE_COLUMN_AS_NULL

Controls how security-sensitive columns are displayed to unauthorized users. If this parameter is set to `YES`, then a `NULL` expression replaces the original column expression in the query and secured columns are hidden from unauthorized users in analyses.

If this parameter is set to `NO`, then when a user attempts to run a report that contains a secured column the user is not authorized to see, an unresolved column error occurs.

The default value is `NO`.

Example: `PROJECT_INACCESSIBLE_COLUMN_AS_NULL = NO;`

A.5.3 IGNORE_LDAP_PWD_EXPIRY_WARNING

Determines whether users can log in even when the LDAP server issues a password expiration warning. Valid values are `YES` and `NO`. Specify `YES` to allow users to log in when the LDAP server issues a password expiration warning, or specify `NO` to reject user logins when the warning is issued. The default value is `NO`.

After user passwords have actually expired in the LDAP server, users cannot log in, regardless of the value of this parameter.

Example: `IGNORE_LDAP_PWD_EXPIRY_WARNING = NO;`

A.5.4 SSL

This parameter, along with the remaining parameters in this section, relate to Secure Socket Layer (SSL) communication between Oracle Business Intelligence components.

The default setting for `SSL` is `NO`.

> **Note:** Most of the SSL parameters in this section are centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended). The centrally managed SSL parameters include:
>
> - SSL
> - SSL_CERTIFICATE_FILE
> - SSL_PRIVATE_KEY_FILE
> - SSL_VERIFY_PEER
> - SSL_CA_CERTIFICATE_FILE
>
> See "SSL Configuration in Oracle Business Intelligence" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for complete information about how to configure SSL between Oracle Business Intelligence components.

A.5.5 SSL_CERTIFICATE_FILE

Specifies the directory path to the certificate file. For components acting as SSL servers, such as Oracle BI Server and Oracle BI Scheduler, this is the Server Certificate filename. For client components, such as Oracle Business Intelligence ODBC Client Data Source, this is the Client Certificate filename.

This parameter is centrally managed.

Example (Server): `SSL_CERTIFICATE_FILE = "servercert.pem";`

Example (Client): `SSL_CERTIFICATE_FILE = "client-cert.pem";`

A.5.6 SSL_PRIVATE_KEY_FILE

Specifies the private key file. For server components, this is the Server Private Key file name. For client components, this is the Client Private Key file name.

This parameter is centrally managed.

Example (Server): `SSL_PRIVATE_KEY_FILE = "serverkey.pem";`

Example (Client): `SSL_PRIVATE_KEY_FILE = "client-key.pem";`

A.5.7 SSL_VERIFY_PEER

Specifies whether the server requires client authentication or not. When set to `YES`, the Oracle Business Intelligence component verifies that the other component in the connection has a valid certificate (that is, mutual authentication). The default value of `NO` permits a connection to any peer.

This parameter is centrally managed.

Example: `SSL_VERIFY_PEER = NO;`

A.5.8 SSL_CA_CERTIFICATE_FILE

Specifies the name and path of the trusted CA Certificate used to verify the server or client certificate when Verify Peer is set to `YES`. Takes effect only when client authentication is required.

This parameter is centrally managed.

Example: `SSL_CA_CERTIFICATE_FILE = "CACertFile";`

A.5.9 SSL_TRUSTED_PEER_DNS

Specifies individual named clients that are allowed to connect by Distinguished Name (DN). The DN identifies the entity that holds the private key that matches the public key of the certificate.

This parameter is *not* centrally managed.

Example: `SSL_TRUSTED_PEER_DNS = "";`

A.5.10 SSL_CERT_VERIFICATION_DEPTH

The depth of the certificate chain. A depth of one means a certificate has to be signed by one of the trusted CAs. A depth of two means the certificate was signed by a CA that was further verified by one of the CAs. The default value is `9`.

This parameter is *not* centrally managed.

Example: `SSL_CERT_VERIFICATION_DEPTH = 9;`

A.5.11 SSL_CIPHER_LIST

A list of permitted cipher suites that the server uses. The default is empty string, which is equivalent to "ALL."

You must set this parameter only when you want to use a cipher suite other than the default choice.

This parameter is *not* centrally managed.

Example: `SSL_CIPHER_LIST = "EXP-RC2-CBC-MD5";`

A.6 Server Section Parameters

The parameters in the Server section define defaults and limits for the Oracle BI Server.

A.6.1 READ_ONLY_MODE

Note: The `READ_ONLY_MODE` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

The **Disallow Online RPD Updates** option on the Performance tab of the Capacity Management page in Fusion Middleware Control corresponds to the `READ_ONLY_MODE` parameter. See Section 7.3.1, "Using Fusion Middleware Control to Disallow RPD Updates" for more information.

Permits or forbids changing Oracle BI repository files when the Administration Tool is in either online or offline mode. The default is `NO`, meaning that repositories can be edited.

When this parameter is set to YES, it prevents the Administration Tool from making any changes to repository files. When the Administration Tool opens the repository, a message informs the user that the repository is read-only. If this parameter is set to NO, then the Administration Tool can make changes to the repository.

Note that even when READ_ONLY_MODE is set to NO, there are still situations when Administration Tool opens repositories in read-only mode. For example, if you open a repository in offline mode, but the Oracle BI Server or another Administration Tool client holds a lock on the repository, then the repository opens in read-only mode. In online mode, a repository might open as read-only if an offline Administration Tool held a lock on the repository at the time the Oracle BI Server started.

In addition, the Administration Tool also opens in read-only mode when Oracle Business Intelligence has been clustered, and the Administration Tool is connected in online mode to a slave node. This occurs because the master node holds a lock on the repository. To avoid this situation when running in a clustered environment, ensure that the Oracle BI Server ODBC DSN that is used by the Administration Tool has been configured to point to the cluster controller rather than to a particular Oracle BI Server.

A.6.2 MAX_SESSION_LIMIT

Specifies the maximum number of concurrent connections that are allowed by the server. When this number is exceeded, the server refuses the connection request.

The limit is 65,535 connections.

Example: `MAX_SESSION_LIMIT = 2000;`

A.6.2.1 About the MAX_SESSION_LIMIT and SERVER_THREAD_RANGE Parameters

The size of the connection pool determines the number of available Oracle BI Server connections and the number of available threads for processing physical queries. A logical query might generate multiple physical queries, each of which could go to different connections.

The Oracle BI Server creates a number of server threads up to the specified maximum using the parameter SERVER_THREAD_RANGE. All the threads available at any time are used to process queries from one or more sessions as needed.

Typically, the number of sessions that is specified by MAX_SESSION_LIMIT is higher than the number of available threads that is specified by SERVER_THREAD_RANGE.

In summary:

- MAX_SESSION_LIMIT specifies the number of sessions that can be connected to the Oracle BI Server, even if inactive. The sessions and the corresponding queries are queued to the threads for processing as they become available.
- The size of the connection pool specifies the number of threads and connections that process physical queries.
- SERVER_THREAD_RANGE specifies the number of threads that process the logical queries, or in other words, the number of queries that can be active in the Oracle BI Server at any time.

A.6.3 MAX_REQUEST_PER_SESSION_LIMIT

Specifies the maximum number of logical requests per session. This is how many open requests there are, per session, at the same time.

The limit is 65,535 logical requests per session.

Note: Usually, individual users have only one open request for each session at the same time. Application programs and Oracle BI Presentation Services, however, typically have multiple requests open at the same time. In general, the default value of 500 should be sufficient for most environments, but this parameter should be tuned based on the application environment and the client tools in use.

Example: `MAX_REQUEST_PER_SESSION_LIMIT = 500;`

A.6.4 SERVER_THREAD_RANGE

For each Oracle BI Server request, `SERVER_THREAD_RANGE` specifies configuration information for thread allocation. The lower number in the range specifies the number of threads that is initially allocated, and the higher number in the range specifies the maximum number of threads to be allocated. The thread pool grows and shrinks in 5-thread increments until the upper or lower bound is reached. If there are fewer threads than sessions, then sessions share the available number of threads on a first come-first served basis.

Although setting both values to the same number maximizes the benefits of thread pooling, there is a cost associated with doing so. If you set the lower boundary and the upper boundary to the same number, then that number of threads is always allocated, which consumes stack space.

Example: `SERVER_THREAD_RANGE = 10-200;`

See Section A.6.2.1, "About the MAX_SESSION_LIMIT and SERVER_THREAD_RANGE Parameters" for related information.

A.6.5 SERVER_THREAD_STACK_SIZE

Specifies the memory stack size that is allocated for each server thread. A value of 0 sets the stack size as 256 KB for each server thread for 32-bit platforms, or 1 MB for 64-bit systems.

The default value is 0. If you change this value, then ensure that the value that you provide is appropriate for the memory resources that are available on the system.

Example: `SERVER_THREAD_STACK_SIZE = 0;`

A.6.6 DB_GATEWAY_THREAD_RANGE

Specifies the minimum and maximum number of threads in the Oracle Business Intelligence Database Gateway thread pool, as per `SERVER_THREAD_RANGE`.

The default value is 40-200.

Example: `DB_GATEWAY_THREAD_RANGE = 40-200;`

A.6.7 DB_GATEWAY_THREAD_STACK_SIZE

Specifies the memory stack size that is allocated for each Oracle Business Intelligence Database Gateway thread. A value of 0 sets the stack size as 256 KB per server thread for 32-bit platforms, or 1 MB for 64-bit systems.

The default value is 0. If you change this value, then ensure that the value that you provide is appropriate for the memory resources that are available on the system.

Example: `DB_GATEWAY_THREAD_STACK_SIZE = 0;`

A.6.8 MAX_EXPANDED_SUBQUERY_PREDICATES

Controls the maximum number of values that can be in an `IN` value list that is populated by a subquery. The default is 8,192 values. The Oracle BI Server generates an error if this limit is exceeded.

Note that there is also a database feature setting called `MAX_ENTRIES_PER_IN_LIST`. This value is set according to how many literals can be supported by the given data source. If this limit is exceeded, then the Oracle BI Server breaks the `IN` list into smaller ones and ORs them together. However, if the original `IN` list is too long, it might exceed the SQL statement length limit for that data source, resulting in a database error or crash. The `MAX_EXPANDED_SUBQUERY_PREDICATES` parameter provides a second limit to ensure that this situation does not occur.

Example: `MAX_EXPANDED_SUBQUERY_PREDICATES = 8192;`

A.6.9 MAX_QUERY_PLAN_CACHE_ENTRIES

Controls the number of cached logical query plans. The query plan cache is an internal performance feature that increases the speed of the query compilation process by caching plans for the most recently used queries.

The default value of this parameter is 1024. Do not raise this value without consulting Oracle Support.

Example: `MAX_QUERY_PLAN_CACHE_ENTRIES = 1024;`

A.6.10 MAX_DRILLDOWN_INFO_CACHE_ENTRIES

Controls the number of cached Action Link drilldown information entries per repository. This increases the speed of computing Action Link information by caching the Action Link information for the most recently used queries.

The default value of this parameter is 1024. Do not raise this value without consulting Oracle Support.

Example: `MAX_DRILLDOWN_INFO_CACHE_ENTRIES = 1024;`

A.6.11 MAX_DRILLDOWN_QUERY_CACHE_ENTRIES

Controls the number of cached Action Link query entries per repository. This increases the speed of drilling down by caching the Action Link drilldown results for the most recently used queries.

The default value of this parameter is 1024. Do not raise this value without consulting Oracle Support.

Example: `MAX_DRILLDOWN_QUERY_CACHE_ENTRIES = 1024;`

A.6.12 INIT_BLOCK_CACHE_ENTRIES

Controls the number of initialization block result sets that are cached with respect to row-wise initialization. The cache key is the fully instantiated initialization block SQL.

The default value is `20`. Because this parameter affects internal operations for localized versions of Oracle Business Intelligence, it is recommended that you do not change this value unless instructed to do so.

Example: `INIT_BLOCK_CACHE_ENTRIES = 20;`

A.6.13 CLIENT_MGMT_THREADS_MAX

Specifies the number of management threads to allocate for managing Oracle BI Server client/server communications. Each client process consumes a management thread. The client/server communication method for Oracle BI Server is TCP/IP.

Because the default value of `5` is typically sufficient for server communications with clients, do not change the value of this parameter.

Example: `CLIENT_MGMT_THREADS_MAX = 5;`

A.6.14 RPC_SERVICE_OR_PORT

Note: The `RPC_SERVICE_OR_PORT` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

The **Port Range From** and **Port Range To** options on the Scalability tab of the Capacity Management page in Fusion Middleware Control override the `RPC_SERVICE_OR_PORT` parameter. See Section 5.5, "Using Fusion Middleware Control to Scale System Components" for more information.

Specifies the IP address and port number on which the Oracle BI Server listens. You can specify an IP address and port number in the form *ip_address*:*port*, or you can specify a port number only.

When you specify an IP address and port number, the Oracle BI Server binds to the specified IP address.

When you specify a port number only, the IP address is set by default to 0.0.0.0, which causes the Oracle BI Server to listen on all IP addresses on that computer.

When you specify an IP address only, the port value defaults to `9703`.

When using the Oracle Business Intelligence ODBC wizard to configure ODBC data sources for the Oracle BI Server, the port number specified in the **Port** field on the Enter Logon Information screen should match the port number specified here. If you change the port number in the configuration file, then ensure that you reconfigure any affected ODBC data sources to use the new port number.

Example1: `RPC_SERVICE_OR_PORT = 9703;`

Example2: `RPC_SERVICE_OR_PORT = 127.0.0.1:9703;`

A.6.15 ENABLE_DB_HINTS

Enables optional hints to be passed along with a SQL statement to an Oracle Database. Database hints are discussed in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

The default value is `YES`.

Example: `ENABLE_DB_HINTS = YES;`

A.6.16 PREVENT_DIVIDE_BY_ZERO

Controls the behavior for when a division by zero occurs. When set to YES, then a NULL value is returned. When set to NO, then the query is terminated and an appropriate error is returned to the user.

Example: PREVENT_DIVIDE_BY_ZERO = YES;

A.6.17 CLUSTER_PARTICIPANT

Note: The CLUSTER_PARTICIPANT parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

When you add an Oracle BI Server instance on the Scalability tab of the Capacity Management page in Fusion Middleware Control, CLUSTER_PARTICIPANT is set to YES for that server. See Section 5.5, "Using Fusion Middleware Control to Scale System Components" for more information.

Specifies whether the Oracle BI Server that is using this configuration file is a member of an Oracle BI Server cluster.

Valid values are YES and NO. The default value is NO.

If the server is to be a member of an Oracle BI Server cluster, then optionally uncomment the parameters "REPOSITORY_PUBLISHING_DIRECTORY" and "REQUIRE_PUBLISHING_DIRECTORY" and supply valid values for them.

When CLUSTER_PARTICIPANT is set to YES, this server must have a valid, configured NQClusterConfig.INI file in the following location:

```
ORACLE_INSTANCE\config\OracleBIClusterControllerComponent\coreapplication_obiccsn
```

For more information, see the information about the NQClusterConfig.INI file in Chapter 6, "Deploying Oracle Business Intelligence for High Availability."

Example: CLUSTER_PARTICIPANT = YES;

A.6.18 REPOSITORY_PUBLISHING_DIRECTORY

Note: The REPOSITORY_PUBLISHING_DIRECTORY parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

The **Shared Location** option on the Repository tab of the Deployment page in Fusion Middleware Control corresponds to the REPOSITORY_PUBLISHING_DIRECTORY parameter. See Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location" for more information.

When the parameter CLUSTER_PARTICIPANT is set to YES, REPOSITORY_PUBLISHING_DIRECTORY specifies the location of the repository publishing directory shared by all Oracle BI Servers participating in the cluster. There is no default value for this parameter.

When a repository is updated in online mode, it is published to this location. All clustered servers examine this location upon startup for any repository changes. This must be a valid location visible to all servers in the cluster, even if you anticipate that no repositories are updated in online mode.

The directory should reside on a shared file system. The directory must be a valid fully qualified directory path name, with double quotes (") surrounding the path name. UNC path names and network mapped drives are allowed only if the service runs under a qualified user account. Do not specify a relative path name, or the Repository subdirectory (located in the Oracle Business Intelligence software installation directory) as the location of the repository publishing directory.

The Oracle BI Server designated as the master server for online repository changes (from the `MASTER_SERVER` parameter in the NQClusterConfig.INI file) must have read and write access to this directory. The Oracle BI Servers in the cluster (from the `SERVERS` parameter in the NQClusterConfig.INI file) must also have read and write access to this directory. All entries must reference the same actual directory, although different names can be specified to accommodate differences in drive mappings.

Examples:

```
REPOSITORY_PUBLISHING_DIRECTORY = "z:\OracleBI\Publish";
```

```
REPOSITORY_PUBLISHING_DIRECTORY = "\\ClusterSrv\Publish";
```

A.6.19 REQUIRE_PUBLISHING_DIRECTORY

Note: The `REQUIRE_PUBLISHING_DIRECTORY` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

The **Share Repository** option on the Repository tab of the Deployment page in Fusion Middleware Control corresponds to the `REQUIRE_PUBLISHING_DIRECTORY` parameter. See Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location" for more information.

When the parameter `CLUSTER_PARTICIPANT` is set to `YES`, `REQUIRE_PUBLISHING_DIRECTORY` specifies that the repository publishing directory (from the parameter `REPOSITORY_PUBLISHING_DIRECTORY`) must be available for this Oracle BI Server to start and join the cluster.

This parameter is commented out by default.

When set to `YES`, if the publishing directory is not available at startup or if an error is encountered while the server is reading any of the files in the directory, an error message is logged in the nqserver.log file and the server shuts down.

To allow the Oracle BI Server to start and join the cluster even if the publishing directory is not available, set this value to `NO`. When set to `NO`, the server joins the cluster and a warning message is logged in the nqserver.log file. Any online repository updates are not reflected in the server's Repository directory (located in the Oracle Business Intelligence software installation directory). This could result in request failures, wrong answers, and other problems. However, this could be useful in situations where online repository editing is done infrequently and the goal is to keep the cluster operational even if some servers have stale repositories.

Example: `REQUIRE_PUBLISHING_DIRECTORY = YES;`

A.6.20 AUTOMATIC_RESTART

Specifies whether the Oracle BI Server should be automatically restarted after a crash. Automatic restart applies only to an Oracle BI Server platform; it does not apply to a clustered Oracle BI Server environment. The default value is `YES`.

Example: `AUTOMATIC_RESTART = YES;`

A.6.21 FMW_SECURITY_SERVICE_URL

Note: The `FMW_SECURITY_SERVICE_URL` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).

Specifies the location where Oracle WebLogic Server is running so that the Oracle BI Server can locate the Oracle Fusion Middleware security service.

Example: `FMW_SECURITY_SERVICE_URL = "http://localhost:9704";`

A.6.22 FMW_SECURITY_SERVICE_MAX_NUMBER_OF_CONNECTIONS

Limits the number of connections from the Oracle BI Server to the Oracle Fusion Middleware security service to avoid overloading the Oracle WebLogic Server with too many connections. Do not change.

Example: `FMW_SECURITY_SERVICE_MAX_NUMBER_OF_CONNECTIONS = 2000;`

A.6.23 FMW_SECURITY_SERVICE_MAX_NUMBER_OF_RETRIES

Specifies the maximum number of times to attempt to connect to the Oracle Fusion Middleware security service.

Example: `FMW_SECURITY_SERVICE_MAX_NUMBER_OF_RETRIES = 0;`

A.6.24 FMW_UPDATE_ROLE_AND_USER_REF_GUIDS

Users are identified by their global unique identifiers (GUIDs), not by their names. Because of this, if you migrate from test to production or change from one identity store to another, then user-based data access security that you configure in the Oracle BI repository might no longer work.

To avoid this situation, you can temporarily set this parameter to `YES` to synchronize (refresh) the GUIDs for users in the repository with the GUIDs for users in the identity store. The default value for this parameter is `NO`.

You do not normally refresh GUIDs in the identity store between test and production environments, because the GUIDs should be identical in both environments. You should synchronize GUIDs only if they differ between test and production environments.

Ensure that you set this parameter back to `NO` after synchronizing the GUIDs to avoid compromising the security of the system.

See "Managing Authentication" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for complete information about the process of refreshing GUIDs. Note that a similar parameter is also needed to refresh GUIDs for Presentation Services.

Example: `FMW_UPDATE_ROLE_AND_USER_REF_GUIDS = NO;`

A.7 Dynamic Library Section Parameters

This section contains one entry for each dynamic link library (DLL) or set of shared objects that is used to make connections to the Oracle BI Server, for both Windows and UNIX systems.

Syntax: *`logical_name`* = *`dynamic_library`*;

Where:

- *`logical_name`*: A logical name for the dynamic link library. These logical names also appear in the Connection Pool dialog.
- *`dynamic_library`*: The name of the associated dynamic library. These libraries are located in:

  ORACLE_HOME/bifoundation/server/bin

> **Caution:** Do not make any changes to this section unless instructed to do so by Oracle Support.

The following are the dynamic link libraries that are shipped with this release:

- ODBC200 = nqsdbgatewayodbc;
- ODBC350 = nqsdbgatewayodbc35;
- OCI8 = nqsdbgatewayoci8;
- OCI8i = nqsdbgatewayoci8i;
- OCI10g = nqsdbgatewayoci10g;
- DB2CLI = nqsdbgatewaydb2cli;
- DB2CLI35 = nqsdbgatewaydb2cli35;
- NQSXML = nqsdbgatewayxml;
- XMLA = nqsdbgatewayxmla;
- ESSBASE = nqsdbgatewayessbasecapi;
- OracleADF = nqsdbgatewayoracleadf;
- OracleADF_HTTP = nqsdbgatewayoracleadf;
- HyperionADM = nqsdbgatewayadm;

A.8 Usage Tracking Section Parameters

The usage tracking parameters define default values for the collection of usage tracking statistics on each logical query submitted to the Oracle BI Server.

Table A–2 shows the names and descriptions of columns that are added to the usage tracking table and to the stand-alone usage tracking repository.

Table A–2 Names and Descriptions of Columns Added to Usage Tracking Table

Name	Data Type	Description	Notes
SAW_DASHBOARD_PG	Varchar(150)	Page within Oracle BI Presentation Services dashboard	Null if not a dashboard request.
PRESENTATION_NAME	Varchar(128)	Name of the Presentation Catalog in Oracle BI Presentation Services	
ERROR_TEXT	Varchar(250)	Error flag and reason text for queries that do not generate a cache entry, from back-end databases	Only applicable if `SUCCESS_FLG` is non-zero. Concatenates multiple messages; the application must parse the column contents.
RUNAS_USER_NAME	Varchar(128)	Impersonated User (the Proxy User that executed the query)	Null if the request is not run as an impersonated user.

For more information about usage tracking, see Chapter 9, "Managing Usage Tracking."

A.8.1 ENABLE

Enables or disables the collection of usage tracking statistics.

Valid values are `YES` and `NO`. The default value is `NO`. When set to `NO`, statistics are not accumulated. When set to `YES`, statistics are accumulated for each logical query.

Example: `ENABLE = NO ;`

A.8.2 DIRECT_INSERT

Specifies whether statistics are inserted directly into a database table or written to a local file.

- When `DIRECT_INSERT` is set to `NO`, data is written to a flat file.
- When `DIRECT_INSERT` is set to `YES`, data is inserted into a table.

> **Note:** This parameter is operative only if the usage tracking parameter `ENABLE` is set to `YES`.

Because direct insertion into a database table is recommended, the default value is `YES`.

Certain other parameters become valid, depending whether `DIRECT_INSERT` is set to `YES` or to `NO`. These parameters are summarized in Table A–3 and described in the following sections.

Table A–3 Valid Parameters for DIRECT_INSERT Settings

DIRECT_ INSERT Setting	Parameters Used	Parameter Setting
NO	STORAGE_DIRECTORY	"*full_directory_path*"
NO	CHECKPOINT_INTERVAL_MINUTES	5
NO	FILE_ROLLOVER_INTERVAL_MINUTES	30
NO	CODE_PAGE	"ANSI"
YES	PHYSICAL_TABLE_NAME	"*Database*"."*Catalog*"."*Schema*". "*Table*" or "*Database*"."*Schema*"."*Table*"
YES	CONNECTION_POOL	"*Database*"."*Connection_Pool*"
YES	BUFFER_SIZE	10 MB
YES	BUFFER_TIME_LIMIT_SECONDS	5
YES	NUM_INSERT_THREADS	5
YES	MAX_INSERTS_PER_TRANSACTION	1

A.8.3 STORAGE_DIRECTORY

Specifies the full path to the directory that is used to store usage tracking log files. The directory listed must be a valid fully qualified, writable directory path name, with double quotes (") surrounding the path name. Specify mapped directories only.

Valid values are any fully qualified path name to an existing, writable directory.

The parameter STORAGE_DIRECTORY is valid only if the parameter DIRECT_INSERT is set to NO. When usage tracking is enabled, but no storage directory is specified, the files are written to the following location:

```
ORACLE_INSTANCE\diagnostics\logs\OracleBIServerComponent\coreapplication_obisn
```

Example: STORAGE_DIRECTORY = "C:\Temp\UsageTracking";

A.8.4 CHECKPOINT_INTERVAL_MINUTES

Specifies how often the usage tracking data is flushed to disk. Setting this interval higher increases the amount of data that might be lost in the event of an abnormal server shutdown. Setting this interval lower incurs additional overhead.

The default is 5 minutes.

> **Note:** When the interval is set to 0, the Oracle BI Server attempts to write usage tracking data to disk with minimal time between attempts. This can negatively affect server performance and is strongly discouraged.

Example: CHECKPOINT_INTERVAL_MINUTES = 5;

A.8.5 FILE_ROLLOVER_INTERVAL_MINUTES

Specifies the time, in minutes, before the current usage tracking log file is closed and a new file is created. For example, if this entry is set to 60 minutes, then 24 usage tracking log files are created each day.

The default is 30 minutes.

When the checkpoint interval equals or exceeds the rollover interval, only the rollover occurs explicitly; the checkpoint occurs implicitly only when the old usage tracking log file is closed.

> **Note:** When the checkpoint interval is set to `0`, the Oracle BI Server attempts to close current usage tracking log files and open new log files with minimal time between attempts. This can negatively affect server performance and result in a large number of usage tracking log files in the storage directory. Setting this interval to `0` is strongly discouraged.

Example: `FILE_ROLLOVER_INTERVAL_MINUTES = 240;`

A.8.6 CODE_PAGE

For multilingual repositories, this specifies the type of output code page to use when writing statistics to disk. Valid values include any valid code page number (such as 1252), and other globally recognized output code page types.

The default value is `ANSI`. The type depends upon the database loader being used. For example, to support multilingual repositories for database loaders that are used by Oracle Database and DB2, specify `UTF8`. Enclose the value in double quotes. USC-2 is currently not supported.

Example: `CODE_PAGE = "ANSI";`

A.8.7 PHYSICAL_TABLE_NAME

Specifies the table in which to insert records that correspond to the query statistics. The table name is the fully qualified name as it appears in the Physical layer of the Administration Tool.

The general structure of this parameter depends on the type of database being used:

- For SQL Server, use the following general structure:

  ```
  PHYSICAL_TABLE_NAME = "Database"."Catalog"."Schema"."Table";
  ```

  Example:

  ```
  PHYSICAL_TABLE_NAME = "OracleBI Usage"."Catalog"."dbo"."S_NQ_
  ACCT";
  ```

  In the preceding example, the structure is as follows:

  - "Oracle BI Usage" represents the database component
  - "Catalog" represents the catalog component
  - "dbo" represents the schema component
  - "S_NQ_ACCT" represents the table name

- For Oracle Database, use the following general structure:

```
PHYSICAL_TABLE_NAME = "Database"."Schema"."Table";
```

Examples:

```
PHYSICAL_TABLE_NAME = "OracleBI Usage"."S_NQ_SCHED"."S_NQ_
ACCT";
```

In the preceding example, the structure is as follows:

- "Oracle BI Usage" represents the database component
- "S_NQ_SCHED" represents the schema component
- "S_NQ_ACCT" represents the table name

A.8.8 CONNECTION_POOL

Specifies the connection pool to use for inserting records into the usage tracking table. This is the fully qualified name as it appears in the Physical layer of the Administration Tool.

Example: `CONNECTION_POOL = "OracleBI Usage"."Connection Pool";`

A.8.9 BUFFER_SIZE

Specifies the amount of memory that is used to temporarily store insert statements. The buffer allows the insert statements to be issued to the usage tracking table independently of the query that produced the statistics to be inserted. When the buffer fills up, then the statistics of subsequent queries are discarded until the insert threads service the buffer entries.

You can specify the size in KB or MB, or enter a number with no suffix to specify bytes.

Example: `BUFFER_SIZE = 10 MB;`

A.8.10 BUFFER_TIME_LIMIT_SECONDS

Specifies the maximum amount of time that an insert statement remains in the buffer before it is issued to the usage tracking table. This time limit ensures that the Oracle BI Server issues the insert statements quickly even during periods of extended quiescence.

Example: `BUFFER_TIME_LIMIT_SECONDS = 5;`

A.8.11 NUM_INSERT_THREADS

Specifies the number of threads that remove insert statements from the buffer and issue them to the usage tracking table. The number of threads should not exceed the total number of threads that are assigned to the connection pool.

Example: `NUM_INSERT_THREADS = 5;`

A.8.12 MAX_INSERTS_PER_TRANSACTION

Specifies the number of records to group as a single transaction when inserting into the usage tracking table. Increasing the number might slightly increase performance, but also increases the possibility of inserts being rejected due to deadlocks in the database.

Example: `MAX_INSERTS_PER_TRANSACTION = 1;`

A.8.13 SUMMARY_STATISTICS_LOGGING

This parameter is reserved for a future release.

A.8.14 SUMMARY_ADVISOR_TABLE_NAME

This parameter is reserved for a future release.

A.9 Query Optimization Flags Section Parameters

There is one parameter in the Query Optimization Flags section. It is a special parameter to override the behavior of the Oracle BI Server in certain situations.

A.9.1 STRONG_DATETIME_TYPE_CHECKING

Use this parameter to relax strong type checking to prevent some date/time data type incompatibilities in queries from being rejected. For example, a query of the form "date/time op string-literal" technically contains a date/time data type incompatibility and would normally be rejected by the Oracle BI Server.

Valid values are `ON` and `OFF`. The default value is `ON`, which means that strong type checking is enabled and queries containing date/time data type incompatibilities are rejected. This is the recommended setting.

To relax the strong type checking, set the value to `NO`. Note that invalid queries or queries with severe date/time incompatibilities are still rejected. Note also that the query could still fail, for example, if the relational database implements a similar strong type checking.

Example: `STRONG_DATETIME_TYPE_CHECKING = ON;`

A.10 MDX Member Name Cache Section Parameters

The parameters in this section are for a cache subsystem that maps between a unique name and the captions of members of all SAP/BW cubes in the repository.

A.10.1 ENABLE

This parameter indicates if the feature is enabled or not.

The default value is `NO` because this only applies to SAP/BW cubes.

A.10.2 DATA_STORAGE_PATH

The path to the location where the cache is persisted. This applies only to a single location.

The number at the end of the entry indicates the storage capacity. When the feature is enabled, the string `<full directory path>` must be replaced with a valid path.

Example:
```
DATA_STORAGE_
PATH = "C:\OracleBI\server\Data\Temp\Cache" 500 MB;
```

A.10.3 MAX_SIZE_PER_USER

The maximum disk space that is allowed for each user for cache entries.

Example: `MAX_SIZE_PER_USER = 100 MB;`

A.10.4 MAX_MEMBER_PER_LEVEL

The maximum number of members in a level that can be persisted to disk.

Example: `MAX_MEMBER_PER_LEVEL = 1000;`

A.10.5 MAX_CACHE_SIZE

The maximum size for each individual cache entry size.

Example: `MAX_CACHE_SIZE = 100 MB;`

A.11 Aggregate Persistence Section Parameters

Oracle Business Intelligence provides an aggregate persistence feature that automates the creation and loading of the aggregate tables and their corresponding Oracle Business Intelligence metadata mappings. The parameters in this section relate to configuring and using the aggregate persistence feature.

A.11.1 AGGREGATE_PREFIX

Specifies the Domain Server Name for aggregate persistence. The prefix must be between 1 and 8 characters long and should not have any special characters ('_' is allowed).

Example: `AGGREGATE_PREFIX = "SA_";`

A.11.2 AGGREGATE_THREAD_POOL_SIZE

Specifies the number of threads to be launched for aggregate persistence. Within each phase, relational loads are executed in separate threads to improve the load performance. The default value is 5.

Example: `AGGREGATE_THREAD_POOL_SIZE = 5;`

A.11.3 AGGREGATE_AW_NAME

This parameter is reserved for a future release.

A.11.4 PREAGGREGATE_AW_CUBE

This parameter is reserved for a future release.

A.12 JavaHost Section Parameters

There is only one parameter in this section. It provides information about the computers where the JavaHost process is running.

A.12.1 JAVAHOST_HOSTNAME_OR_IP_ADDRESSES

> **Note:** The `JAVAHOST_HOSTNAME_OR_IP_ADDRESS` parameter is centrally managed by Fusion Middleware Control and cannot be changed by manually editing NQSConfig.INI, unless all configuration through Fusion Middleware Control has been disabled (not recommended).
>
> The host information and **Port Range From** and **Port Range To** options on the Scalability tab of the Capacity Management page in Fusion Middleware Control override the `JAVAHOST_HOSTNAME_OR_IP_ADDRESS` parameter. See Section 5.5, "Using Fusion Middleware Control to Scale System Components" for more information.

This parameter provides information about JavaHost connectivity. The default port value is `9810`.

Syntax: `JAVAHOST_HOSTNAME_OR_IP_ADDRESS = "`*`host_name1`*`:`*`port1`*`",`*`host_name2`*`:`*`port2`*`;`

Example: `JAVAHOST_HOSTNAME_OR_IP_ADDRESS = "MYHOST:9810";`

B

Exchanging Metadata with Databases to Enhance Query Performance

If your organization has installed either Oracle Database or IBM DB2, then you can use these databases to enhance the data warehouse performance and functionality of queries that run on the Oracle BI Server.

This appendix contains the following topics:

- Section B.1, "About Exchanging Metadata with Databases"
- Section B.2, "Generating the Import File"
- Section B.3, "Using Materialized Views in the Oracle Database with Oracle Business Intelligence"
- Section B.4, "Using IBM DB2 Cube Views with Oracle Business Intelligence"

B.1 About Exchanging Metadata with Databases

By exchanging Oracle Business Intelligence metadata from the Oracle BI Server with your Oracle Database or IBM DB2 database, you enable the database to accelerate the performance of data warehouse queries.

You use the Oracle BI Server utility sametaexport to exchange the metadata. When you run this utility to generate cube views for DB2, the utility is called the DB2 Cube Views Generator. When you run this utility to generate metadata for Oracle Database, the utility is called the Oracle Database Metadata Generator.

The Oracle BI Server export utility works with the following tools:

- In the Oracle Database, the SQL Access Advisor creates materialized views and index recommendations on optimizing performance. Note that in database releases prior to 10g, this feature is called the Oracle Database Summary Advisor.
- In the IBM DB2 database, IBM DB2 Cube Views creates materialized query tables (MQTs).

The sametaexport utility generates the information necessary for the SQL Access Advisor or IBM DB2 Cube Views tool to preaggregate the relational data and improve query performance.

B.2 Generating the Import File

Both the Oracle Database Metadata Generator and the DB2 Cube Views Generator create the files that are needed to import metadata from the Oracle BI Server into the SQL Access Advisor or an IBM DB2 database.

This section contains the following topics that are common to the two generators:

- Section B.2.1, "Running the Generator"
- Section B.2.2, "About the Metadata Input File"
- Section B.2.3, "About the Output Files"
- Section B.2.4, "Troubleshooting Errors from the Generator"
- Section B.2.5, "Metadata Conversion Rules and Error Messages"

B.2.1 Running the Generator

The Oracle Database Metadata Generator and the DB2 Cube Views Generator are invoked from the command line or embedded in a batch file. The command-line executable is named sametaexport.

Before running the utility, you must first run bi-init.cmd (or bi-init.sh on UNIX) to launch a command prompt or shell window that is initialized to your Oracle instance. You can find this utility in:

ORACLE_INSTANCE/bifoundation/OracleBIApplication/coreapplication/setup

Then, run sametaexport with the desired options, as follows:

```
sametaexport -r "PathAndRepositoryFileName" [-p repository_password]
-f "InputFileNameAndPath" [options]
```

Table B–1 contains descriptions of the parameters in the command-line executable file.

Table B–1 Parameters for sametaexport

Parameter	Definition	Additional Information
-r	Repository file name and full path	Quotation marks are required for the file name and path only if the file path is in long format or has spaces. Use the full path if the file is not in the current directory.
-p	Repository password	The password for the given repository. The password argument is optional. If you do not provide a password argument, you are prompted to enter a password when you run the command. To minimize the risk of security breaches, Oracle recommends that you do not provide a password argument either on the command line or in scripts. Note that the password argument is supported for backward compatibility only, and will be removed in a future release.
-f	Input file name and full path	Quotation marks are required for the file name and path only if the file path is in long format or has spaces. Use the full path if the file is not in the current directory. You specify input files so that you do not have to type all the required information at the command line, and so that you can type international characters. See Section B.2.2, "About the Metadata Input File" for more information.

You can include some additional parameters in the input file or at the command line to change various defaults for the Oracle Database Metadata Generator and the DB2 Cube Views Generator. Parameters specified in the input file take precedence over parameters specified at the command line. You must include these parameters only if you want to change the default values.

Table B–2 and Table B–3 describe these optional parameters.

Table B–2 Optional Parameters and Defaults for the Oracle Database Metadata Generator

Parameter Definition	Additional Information	Input File Usage Example	Command Line Usage Example
Use schema name from RPD	When set to YES, the table schema names are used as they are used in the repository. The default value is YES.	USE_SCHEMA_NAME_FROM_RPD = NO	-schemafrom rpd NO
Default schema name	The default schema name is used as the table schema name if the value of -schemafromrpd is set to NO, or if the repository schema name cannot be determined. The default value is SIEBEL.	DEFAULT_SCHEMA_NAME = ORACLE	-defaultschema ORACLE
Oracle schema name	The metadata from Oracle Database Metadata Generator is created under this schema. The default value is SIEBEL.	ORA_DIM_SCHEMA_NAME = ORACLE	-orclschema ORACLE
Logging enabled	Indicates whether to keep a log of the metadata export process. Valid values are ON, OFF, and DEBUG. The default value is ON.	LOGGING = DEBUG	-logging DEBUG
Log file name	The path to the log file. If you provide an invalid path, an error occurs. If you do not provide this parameter, the default log file path is used. The default path is: *ORACLE_INSTANCE*\diagnostics\logs\OracleBIServerComponent\coreapplication_obis*n*\OraDimExp.log	LOG_FILE_NAME = C:\bea_default\instances\instance1\diagnostics\logs\generator\logfile.log	-logfile C:\bea_default\instances\instance1\diagnostics\logs\generator\logfile.log

Table B–3 Optional Parameters and Defaults for the DB2 Cube Views Generator

Parameter Definition	Additional Information	Input File Usage Example	Command Line Usage Example
Distinct count supported	When set to YES, allows measure containing the DISTINCT_COUNT aggregation to be exported. The recommended setting and default value is NO.	DISTINCT_COUNT_SUPPORTED = YES	-distinct YES
Statistical functions supported	When set to YES, allows measures containing the aggregation STDDEV to be exported. The recommended setting and default value is NO.	STATISTICAL_FUNCTIONS_SUPPORTED = YES	-stat YES
Use schema name	When set to YES, the Cube Views metadata attributes have columns from tables under a schema name, which are then specified in the parameters. When set to NO, the schema names for these tables are empty. The default value is YES.	USE_SCHEMA_NAME = NO	-useschema NO
Use schema name from RPD	When set to YES, the table schema names are used as they are used in the repository. The default value is YES.	USE_SCHEMA_NAME_FROM_RPD = NO	-schemafromrpd NO

Table B–3 (Cont.) Optional Parameters and Defaults for the DB2 Cube Views Generator

Parameter Definition	Additional Information	Input File Usage Example	Command Line Usage Example
Default schema name	The default schema name is used as the table schema name if the value of `-schemafromrpd` is set to `NO`, or if the repository schema name cannot be determined. The default value is `SIEBEL`.	DEFAULT_SCHEMA_NAME = ORACLE	-defaultschema ORACLE
Cube views schema name	The name of the schema under which the Cube Views metadata is created. The default value is `SIEBEL`.	CUBE_VIEWS_SCHEMA_NAME = ORACLE	-cubeschema ORACLE
Log file name	The path to the log file. If you provide an invalid path, an error occurs. If you do not provide this parameter, the default log file path is used. The default path is: *`ORACLE_INSTANCE`*`\diagnostics\logs\OracleBIServerComponent\coreapplication_obis`*`n`*`\CubeViews.log`	LOG_FILE_NAME = C:\bea_default\instances\instance1\diagnostics\logs\generator\logfile.log	-logfile C:\bea_default\instances\instance1\diagnostics\logs\generator\logfile.log
Log failures	When set to `YES`, the log file lists the metadata that was invalidated under a certain rule. The default value is `YES`.	LOG_FAILURES = NO	-logfail NO
Log success	When set to `YES`, the log file lists the metadata that has been checked under each rule and has passed the check. The default value is `NO`.	LOG_SUCCESS = YES	-logsuccess YES

B.2.2 About the Metadata Input File

The input file is a text file that contains the parameters that are described in Table B–4.

Table B–4 Cube Metadata Input File Parameters

Input File Name	Description
BUSINESS_MODEL	The name of the business model in the logical layer of the Oracle Business Intelligence repository that contains the metadata to export. If the business model is not found in the repository, then an error message is displayed. You can only specify one business model name in the input file. To generate metadata for multiple business models, create another input file and run the Oracle Database Metadata Generator or DB2 Cube Views Generator again.
PHYSICAL_DATABASE	The name of the database in the physical layer of the Oracle Business Intelligence repository that contains the metadata to export. When the business model derives from multiple databases, then it eliminates metadata from all databases other than the one specified here. When the physical database is not found in the repository, an error message is displayed.
RUN_AS_USER	The user name of the database user whose visibility must be duplicated for the metadata export. This parameter cannot be empty. This user must exist as a user reference in the repository.

Table B–4 (Cont.) Cube Metadata Input File Parameters

Input File Name	Description
OUTPUT_FOLDER	The full path and file name of the folder to which the SQL file will be written. If the folder does not exist when you run the Oracle Database Metadata Generator, then it will be created. See Section B.2.3, "About the Output Files" for more information.

The following text shows a sample metadata input file:

```
BUSINESS_MODEL = "1 - Sample App"
PHYSICAL_DATABASE = "1 - Sample App Data"
RUN_AS_USER = "Administrator"
OUTPUT_FOLDER = "C:\OracleBI"
```

B.2.3 About the Output Files

Each Generator creates different types of output files, as described in the following list:

- **Oracle Database Metadata Generator:** Generates a SQL file that is encoded in UTF-8 and stored in the specified output folder. The file name is based on the name of the business model you specified in the input file, such as my_business_model.sql.
- **DB2 Cube Views Generator:** Generates the following files in the specified output folder:
  - XML file (encoded in UTF8). One XML file is created for the specified business model. It contains all objects that were converted to cubes. Additionally, objects in the repository will be mapped to similar objects in the IBM Cube Views metadata. See Section B.2.5.2, "Conversion Rules for IBM DB2 Databases" for a list of objects that will not be converted.

    The name of the XML file matches the business model name, without spaces, followed by the XML extension (for example, SalesResults.xml).
  - A SQL file that contains the alias generation DLL. A SQL file is created for the specified business model only if aliases exist in the physical layer databases that are referenced in the business model. The alias file contains SQL commands that will create the aliases in the DB2 database. The name of the SQL file matches the business model name, without spaces, followed by the SQL extension (for example, SalesResults-alias.sql).

B.2.4 Troubleshooting Errors from the Generator

Error messages indicate that the Generator was unable to complete some or all of its tasks. After starting the Generator, you might observe the following error messages:

- `Unable to write to Log file:` *`log_file_name`*`.`

  The log file specified in the input file or at the command line might contain the wrong path, the user might not have write permissions to that folder, or the disk could be out-of-space.
- `Run_as_user,` *`user_name`*`, is invalid.`

  The user name is incorrect.
- `Repository,` *`repository_name.rpd`*`, is invalid or corrupt.`

  The repository name might be incorrect, it might not exist in the given path, or the user might not have permission to read it.

- `Physical Database, *database_name*, is invalid.`

  The physical database name does not match a valid physical database object in the repository.

- `Business Model, *model_name*, is invalid.`

  The business model name does not match a valid business model object in the repository.

- `Authentication information provided is invalid.`

  The repository password provided at the command line is incorrect.

- `Path: "*path_name*" is invalid.`

  The path or file name is incorrect, or the current user does not have read access.

B.2.5 Metadata Conversion Rules and Error Messages

When the Generator creates the output files, it also maps the metadata objects in the Oracle Business Intelligence repository to similar objects in the metadata of the Oracle Database or the IBM DB2 database.

This section explains the rules used to identify Oracle Business Intelligence metadata that cannot be translated (converted) into either SQL or XML format. These rules are necessary because Oracle Database and IBM Cube Views do not support some of the metadata constructs that are allowed by Oracle Business Intelligence.

Dimensional metadata in the SQL or XML file will be generated at the logical fact table source level. If a logical fact table source has an invalid logical dimension table source, then the logical dimension table source will be invalidated. If the logical fact table source is invalid, then all the logical dimension table sources that are mapped to it will also be invalidated. Invalid Oracle Business Intelligence repository metadata elements will not be converted to cubes in the SQL or XML file.

When a rule is violated, the Generator writes the error messages and the metadata that violated the rule to the log file.

B.2.5.1 Conversion Rules for Oracle Databases

The following list provides the rules for converting Oracle Business Intelligence metadata into objects in the Oracle Database:

- Attributes that contain expressions in the logical table cannot be exported.
- Tables joined using complex joins are not considered.
- Tables that are opaque views are not considered.
- Columns used as part of a key in one level cannot be used as part of another level key.

  Oracle Database prohibits the use of columns as keys in multiple levels. This prohibition requires the Oracle Database Metadata Generator to eliminate one of the two joins, usually the join that is encountered first. Therefore, the other joins are lost, which prevents them from being exported.

B.2.5.2 Conversion Rules for IBM DB2 Databases

Table B–5 lists the rules used to validate Oracle Business Intelligence repository metadata elements, error messages that are written to the log file if the rule is violated, and an explanation of what caused the rule violation. The error messages help you determine why a particular Oracle Business Intelligence metadata object was not exported to the XML file.

Table B–5 Validation Rules for Metadata Elements

Rule	Message	Explanation
ComplexJoinFactsRule	[Fact Logical Table Source]Complex Physical Joins not supported %qn has a complex Join %qn between Physical Tables %qn and %qn	If the physical fact tables are connected through complex joins, then the join is not supported. A complex join is defined as any join between two tables that do not have a foreign key relationship.
ComplexJoinDimsRule	[Dimension Logical Table Source]Complex Physical Joins not supported %qn has a complex Join %qn between Physical Tables %qn and %qn	If the dimension physical tables are connected through a complex join, then that join is not supported.
ComplexJoinFactDimRule	[Fact Logical Table Source -> Dimension Logical Table Source] Complex Physical Joins not supported. %qn has a complex Join %qn between Physical Tables %qn and %qn.	If a dimension physical table and a fact physical table are connected through a complex join, then that join is not supported and the dimension table source is invalidated.
OpaqueViewFactRule	[Fact Logical table Source] Physical SQL Select Statements not supported. %qn uses the SQL Select Statement %qn.	When the physical fact table is generated by a SQL select statement, the logical fact table source that contains the table is invalidated. All logical dimension table sources connected to this logical fact table source are also invalidated. This construct allows subquery processing.
OpaqueViewDimRule	[Dimension Logical table Source] Physical SQL Select Statements not supported. %qn uses the SQL Select Statement %qn.	When a physical dimension table is generated by a SQL select statement, the logical dimension table source containing that table is invalidated.
OuterJoinFactRule	[Fact Logical Table Source] Physical Outer Joins not supported. %qn has an outer join %qn between physical tables %qn and %qn.	If the logical fact table source has an outer join mapping, then that logical fact table source is invalidated and all logical dimension table sources mapped to this source will also be invalidated.
OuterJoinDimRule	[Dimension Logical Table Source] Physical Outer Joins not supported. %qn has an outer join %qn between physical tables %qn and %qn.	If the logical dimension table source has an outer join mapping, then that logical dimension table source is invalidated.
WhereClauseFactRule	[Fact Logical Table Source] WHERE clauses are not supported. %qn has a where condition %s.	If the fact table source uses a `WHERE` clause to filter the data that is loaded, then this table source is invalidated.
WhereClauseDimRule	[Dimension Logical Table Source] WHERE clauses are not supported. %qn has a where condition %s.	If the dimension table source uses a `WHERE` clause to filter the data that is loaded, then this table source is invalidated.

Table B–5 (Cont.) Validation Rules for Metadata Elements

Rule	Message	Explanation
TwoJoinFactDimRule	[Fact Logical Table Source -> Dimension Logical Table Source] Multiple Joins between sources not supported. %qn and %qn have at least the following joins : %qn, %qn.	If a physical fact table is mapped to two dimension tables from the same dimension source (if the fact table is not exclusively mapped to the most detailed table in the table source), then the dimension table source is invalidated.
HiddenManyManyRule	[Fact Logical Table Source -> Dimension Logical Table Source] Join between (physical or logical?) fact and dimension is not on the most detailed table. %qn between %qn and %qn is not on the most detailed table %qn {Join name, facttable, dimtable).	This is related to the TwoJoinFactDimRule. If the fact table is joined to a dimension table that is not the most detailed table in the table source, then the dimension table source is invalidated.
ComplexMeasureRule	[Column] Complex Aggregation Rules not supported. %qn uses an aggregation rule of %s which is not supported.	The supported aggregations are typically `SUM`, `COUNT`, `AVG`, `MIN`, `MAX`, `STDDEV`, `COUNTDISTINCT`, and `COUNT`.
CountDistMeasureRule	[Column] COUNT-DISTINCT Aggregation Rule not supported. %qn uses an aggregation rule of %s which is not supported.	`COUNTDISTINCT` aggregation is not supported for this particular column.
InvalidColumnLevelRule	[Level] Some columns that are part of the Primary Level Key are invalid. %qn has %qn as part of its primary key, when %qn has already been marked invalid.	The level key for this level has one or more columns that are invalid.
VariableBasedColumnRule	[Logical Table Source -> Column] Column uses a Variable in the Expression Column %qn uses a variable in its mapping.	The logical column uses repository and session variables in the expression.
OneFactToManyDimRule	[Fact Logical Table Source -> Dimension Logical Table Source] There must be a unique join path between the most detailed tables in the (logical or physical?) fact and the dimension. No join paths found between %qn and %qn (both physical table names). Found at least the following join paths: (%qn->%qn....), (%qn->%qn....)	Same as in TwoJoinFactDimRule or HiddenManyManyRule.
ManyMDTinFactRule	[Fact Logical Table Source] Fact Logical Table Source must have a unique most detailed table. %qn has at least the following most detailed tables : %qn,%qn.	A fact that has more than one table that is the most detailed table.
NoMeasureFactRule	[Fact Logical Table Source] Fact Logical Table Source does not have any Measures. %qn does not have any deployable measures.	A fact table does not have any measures because all the measures have been invalidated.

Table B–5 (Cont.) Validation Rules for Metadata Elements

Rule	Message	Explanation
NoInActiveFactRule	[Fact Logical Table Source] Fact Logical Table Source is not marked Active.	A fact source is not active.
NoInActiveDimRule	[Dimension Logical Table Source] Dimension Logical Table Source is not marked Active.	A dimension source is not active.
NoAttributeInFactRule	[Fact Logical Table Source -> Column] Attribute found in Fact. %qn in a fact source %qn does not have an aggregation rule.	No attributes in the fact source.
NoMeasureInDimRule	[Dimension Logical Table Source -> Column] Measure found in Dimension. %qn in a dimension source %qn has an aggregation rule.	No measures in the dimension source.
VisibleColumnsAttrRule	[Column] -> The run_as_user does not have visibility to this Logical Column. %qn is not accessible to the run_as_user %qn due to visibility rules.	A column does not have visibility for this user.
VisibleColumnsMeasRule	[Column] -> The run_as_user does not have visibility to this Logical Column. %qn is not accessible to the run_as_user %qn due to visibility rules.	A column does not have visibility for this user.
MultiplePrimaryKeysDimRule	[Dimension Logical Table Source] A Join uses an alternate key in the Dimension Logical Table Source. %qn between %qn and %qn in %qn uses the alternate key %qn.	A dimension physical table can contain only one primary key. It is joined to another dimension physical table using a different unique key and that join is invalid. IBM Cube Views does not accept any unique keys to be used for foreign joins and always requires the primary key.
MultiplePrimaryKeysFactRule	[Dimension Logical Table Source] A Join uses an alternate key in the Dimension Logical Table Source. %qn between %qn and %qn in %qn uses the alternate key %qn.	A fact physical table can contain only one primary key. It is joined to another fact physical table using a different unique key and that join is invalid. IBM Cube Views does not accept any unique keys to be used for foreign joins and always requires the primary key.
MultiplePrimaryKeysFactDimRule	[Fact Logical Table Source -> Dim Logical Table Source] A Join uses an alternate key between the Logical Table sources. %qn between %qn and %qn for sources %qn and %qn uses the alternate key %qn.	A fact physical table can contain only one primary key. It is joined to a dimension physical table using a different unique key and is invalid. IBM Cube Views does not accept any unique keys to be used for foreign joins and always requires the primary key.
NotDB2ExpressionAttrRule	[Dimension Logical Table Source -> Column] The Column contains an Expression not supported. %qn has expression %s which is not supported.	The attribute contains an expression not supported by IBM Cube Views. This includes metadata expressions that use DateTime functions (for example, `CURRENT_DATE`).

Table B–5 (Cont.) Validation Rules for Metadata Elements

Rule	Message	Explanation
NotDB2ExpressionMeasRule	[Fact Logical Table Source -> Column] The Column contains an Expression not supported. %qn has expression %s which is not supported.	A measure contains an expression not supported by IBM Cube Views. This includes metadata expressions that use DateTime functions (for example,. `CURRENT_DATE`).
NoAttributeDimRule	[Dimension Logical Table Source] Dimension Logical Table Source does not have any attributes visible to the run_as_user. %qn can not be queried by user %qn since none of its attributes are visible.	A dimension does not have any attributes.

B.3 Using Materialized Views in the Oracle Database with Oracle Business Intelligence

This section explains how to export metadata from Oracle Business Intelligence into the SQL Access Advisor and create materialized views using the Oracle Database Metadata Generator.

This section contains the following topics:

- Section B.3.1, "About Using the SQL Access Advisor with Materialized Views"
- Section B.3.2, "Deploying Metadata for Oracle Database"

B.3.1 About Using the SQL Access Advisor with Materialized Views

This feature enhances the data warehouse performance and functionality of a database. It enables the SQL Access Advisor to store metadata about the logical relationships of the data that resides in the database. Additionally, it accelerates data warehouse queries by using more efficient Oracle materialized views. These materialized views preaggregate the relational data and improve query performance. Once the metadata is stored in the SQL Access Advisor, the database administrator can optimize the database objects and improve query performance.

When processing queries, Oracle Database routes queries to tables that hold materialized views when possible. Because these tables of materialized views are smaller than the underlying base tables and the data has been pre aggregated, the queries that are rerouted to them might run faster.

Oracle Database Metadata Generator works as a metadata bridge to convert the Oracle Business Intelligence proprietary metadata into a SQL file that contains PL/SQL commands to generate dimensions in the SQL Access Advisor. After converting metadata into a SQL file, you use a tool such as SQL*Plus to import the translated metadata into the SQL Access Advisor and store it in metadata catalog tables. After importing the metadata, you create materialized views, which are used by to optimize incoming application queries.

You can use this feature with Oracle Database 9*i* and higher. For information about platform compatibility, see Section 1.8, "System Requirements and Certification".

Note that in database releases prior to 10g, the SQL Access Advisor was called the Oracle Database Summary Advisor and was documented in *Oracle9i Data Warehousing Guide*.

B.3.2 Deploying Metadata for Oracle Database

Become familiar with the Oracle Database and its tools before attempting to deploy metadata in the Oracle Database. For more information, see "SQL Access Advisor" in *Oracle Database Performance Tuning Guide*.

Ensure that you complete the steps in Section B.2.1, "Running the Generator" before deploying metadata. To deploy cube metadata, perform the tasks described in the following sections:

- Section B.3.2.1, "Executing the SQL File for Oracle Database"
- Section B.3.2.2, "Defining Constraints for the Existence of Joins"
- Section B.3.2.3, "Creating the Query Workload"
- Section B.3.2.4, "Creating Materialized Views"

B.3.2.1 Executing the SQL File for Oracle Database

Before executing the SQL file for importing into the SQL Access Advisor, ensure that you are familiar with Oracle Database import tools. See the Oracle Database documentation set for information.

Use a tool such as SQL*Plus to execute the SQL file that the Oracle Database Metadata Generator generated. You might see error messages if the dimensions already exist or if the database schema differs from that in the RPD file. When the script executes successfully, you can see the dimensions that were created by using the database web console or the Oracle Enterprise Manager Database Control. In the Oracle Enterprise Manager Database Control, expand the following nodes: Network, Databases, database-name, Warehouse, Summary Management, Dimensions, System.

After you execute the SQL file, be aware of the following:

- No incremental metadata changes are allowed. Schema changes require that you manually delete cube model metadata in the Oracle Database and convert the Oracle Business Intelligence metadata again. For example, if you must make a change to a dimension in a cube in the Oracle BI repository, you must delete the cube model in the Oracle Database, regenerate the SQL file from the Oracle BI repository, and import it into the SQL Access Advisor.
- You cannot delete metadata using the Oracle Database Metadata Generator. Instead, you must manually delete the cube model using the Oracle Enterprise Manager Database Control.

B.3.2.2 Defining Constraints for the Existence of Joins

For more information on this topic, see the Oracle Database documentation set.

You must ensure that Oracle Database knows about the joins between the dimension tables and the fact tables. To do so, you create constraints in SQL*Plus or the Oracle Enterprise Manager Database Control. In the Oracle Enterprise Manager Database Control, you select the table on which you must create a constraint, then select the Constraint tab.

You create a different type of constraint for each kind of table, as follows:

- For dimension tables, create a `UNIQUE` key constraint.
- For fact tables, create a `FOREIGN` key constraint and specify the referenced schema and referenced table. In the Constraint Definition area, include the foreign key columns in the fact table and the corresponding unique keys in the dimension

table. An attempt to create a foreign key on a fact table can fail if the foreign key column data does not match the unique key column data on the dimension table.

B.3.2.3 Creating the Query Workload

See the Oracle Database documentation set for detailed information about creating the query workload.

A query workload is a sample set of physical queries to optimize. Before you create the workload, you generate a Trace file with information on the slowest-running queries.

To generate the Trace file:

You can generate the Trace file of the slowest-running queries using a tool that is appropriate to your database version, as described in the following list:

- **Usage Tracking:** Use this capability in Oracle Business Intelligence to log queries and how long they take to run. Long-running Oracle Business Intelligence queries can then be executed as a script and used with the Trace feature in the Oracle Database to capture the Oracle Database SQL code for these queries.
- **Oracle Database Trace:** Use this tool to identify the slowest physical query. You can enable the Trace feature either within Oracle Enterprise Manager Database Control or by entering SQL commands with the `DBMS_MONITOR` package. Once you enable the Trace feature, you use a script to create a Trace file to capture the SQL code for queries in a query workload table.
- **Oracle Enterprise Manager:** Use this tool to track slow-running queries.

> **Note:** The capabilities that are described in the following sections are available in Oracle Database, rather than as part of Oracle Business Intelligence.

To analyze the information in the Trace file:

1. Use the following guidelines when reviewing the Trace file:
   - When you have traced many statements at once, such as in batch processes, quickly discard any statements that have acceptable query execution times. Focus on those statements that take the longest times to execute.
   - Check the Query column for block visits for read consistency, including all query and subquery processing. Inefficient statements are often associated with a large number of block visits. The Current column indicates visits not related to read consistency, including segment headers and blocks that will be updated.
   - Check the Disk column for the number of blocks that were read from disk. Because disk reads are slower than memory reads, the value will likely be significantly lower than the sum of the Query and Current columns. If it is not, check for issues with the buffer cache.
   - Locking problems and inefficient PL/SQL loops can lead to high CPU time values even when the number of block visits is low.
   - Watch for multiple parse calls for a single statement, because this indicates a library cache issue.
2. After identifying the problem statements in the file, check the execution plan to learn why each problem statement occurred.

To load queries into the workload:

- After you use the Trace utility to learn the names of the slowest physical queries, insert them into the `USER_WORKLOAD` table.

  Table B–6 describes the columns of the `USER_WORKLOAD` table.

- Use `INSERT` statements to populate the `QUERY` column with the SQL statements for the slowest physical queries and the `OWNER` column with the appropriate owner names.

Table B–6 Columns in USER_WORKLOAD Table

Column	Data Type	Require d	Description
QUERY	Any LONG or VARCHAR type (all character types)	YES	SQL statement for the query.
OWNER	VARCHAR2 (30)	YES	User who last executed the query.
APPLICATION	VARCHAR2 (30)	NO	Application name for the query.
FREQUENCY	NUMBER	NO	Number of times that the query was executed.
LASTUSE	DATE	NO	Last date on which the query was executed.
PRIORITY	NUMBER	NO	User-supplied ranking of the query.
RESPONSETIME	NUMBER	NO	Execution time of the query in seconds.
RESULTSIZE	NUMBER	NO	Total number of bytes that the query selected.
SQL_ADDR	NUMBER	NO	Cache address of the query.
SQL_HASH	NUMBER	NO	Cache hash value of the query.

B.3.2.4 Creating Materialized Views

After you populate the query workload table, use the appropriate tool for the Oracle Database version to create materialized views. In Oracle Database 10*g*, use the SQL Access Advisor in the Oracle Enterprise Manager Database Control and specify the query workload table that you created.

The SQL Access Advisor generates recommendations on improving the performance of the fact tables that you specify. The SQL Access Advisor displays the SQL code with which it will create the appropriate materialized views. Before indicating that the SQL Access Advisor should create the materialized views, review the following tips:

- The creation of a materialized view can fail if the SQL code includes a `CAST` statement.
- Ensure that the `CREATE MATERIALIZED VIEW` statement does not specify the same query that you provided as a workload table. If the statement does specify the same query, then the materialized views will likely not reflect the true performance gain. However, if the query is executed frequently, then creating a materialized view might still be worthwhile.
- Add a forward slash (/) to the end of the `CREATE MATERIALIZED VIEW` statement after the SQL statement. Otherwise, the SQL*Plus worksheet will not recognize it as a valid statement.

Note: The SQL Access Advisor can also help determine appropriate indexing schemes.

B.4 Using IBM DB2 Cube Views with Oracle Business Intelligence

This section explains how to export metadata from Oracle Business Intelligence into IBM DB2 using the DB2 Cube Views Generator.

This section contains the following topics:

- Section B.4.1, "About Using IBM DB2 Cube Views with Oracle Business Intelligence"
- Section B.4.2, "Deploying Cube Metadata"

B.4.1 About Using IBM DB2 Cube Views with Oracle Business Intelligence

The term IBM DB2 Cube Views is a registered trademark of IBM. For information about platform compatibility, see Section 1.8, "System Requirements and Certification".

This feature enhances the data warehouse performance and functionality of a database. It enables the DB2 database to store metadata about the logical relationships of the data residing in the database. Additionally, it accelerates data warehouse queries by using more efficient DB2 materialized query tables (MQTs). These MQTs preaggregate the relational data and improve query performance.

When processing queries, the DB2 Query Rewrite functionality routes queries to the MQTs when possible. Because these tables are smaller than the underlying base tables and the data has been pre aggregated, the queries that are rerouted to them might run faster.

DB2 Cube Views Generator works as a metadata bridge to convert the Oracle Business Intelligence proprietary metadata into an IBM Cube Views XML file. After converting metadata into an XML file, you use IBM Cube Views to import the translated metadata into the DB2 database and store it in IBM Cube Views metadata catalog tables. After importing the metadata, you use the IBM Optimization Advisor to generate scripts to create materialized query tables (MQT) and their indexes. The deployed MQTs are used by the DB2 Query Reroute Engine to optimize incoming application queries.

DB2 provides an API (implemented as a stored procedure) that passes XML documents as arguments to create, modify, delete, or read the metadata objects. For more information about IBM Cube Views, see the IBM DB2 documentation.

B.4.2 Deploying Cube Metadata

The alias-SQL file generated by the DB2 Cube Views Generator should be executed before importing the XML file. The XML file generated by the DB2 Cube Views Generator contains the cube metadata in XML format. After importing the XML file into your DB2 database, you must create materialized query tables.

Note: It is strongly recommended that you become familiar with IBM Cube Views and its tools before attempting to import the XML file. For more information, see the IBM documentation.

Ensure that you complete the steps in Section B.2.1, "Running the Generator" before deploying metadata. To deploy cube metadata, perform the tasks described in the following sections:

- Section B.4.2.1, "Executing the Alias-SQL File for IBM Cube Views"
- Section B.4.2.2, "Importing the XML File"
- Section B.4.2.3, "Guidelines for Creating Materialized Query Tables (MQTs)"

B.4.2.1 Executing the Alias-SQL File for IBM Cube Views

You must execute the alias-SQL file before you import the XML file into your DB2 database. For more information, see the IBM documentation.

The alias-SQL file that is generated by the DB2 Cube Views Generator must be executed by a SQL client on the database where the data warehouse is located. When executed, it creates aliases (synonyms) for tables in the database.

B.4.2.2 Importing the XML File

After you execute the alias-SQL file, you can import the XML file into the database. For more information, see the IBM documentation.

> **Note:** It is strongly recommended that you become familiar with IBM Cube Views and its tools before attempting to import the XML file. For more information, see the IBM documentation.

You can import this file using the following IBM tools:

- **IBM OLAP Center (recommended).** For more information, see Section B.4.2.2.1, "Guidelines for Importing the XML File Using the IBM OLAP Center" and the IBM documentation.
- **IBM command-line client utility (db2mdapiclient.exe).** IBM ships this utility with DB2. For more information about using the command-line client utility, see the IBM documentation.
- **IBM DB2 Stored Procedure.** IBM Cube Views provides a SQL-based and XML-based application programming interface (API) that you can use to run a single stored procedure to create, modify, and retrieve metadata objects. For more information, see the IBM documentation.

B.4.2.2.1 Guidelines for Importing the XML File Using the IBM OLAP Center Using the IBM OLAP Center, you can import cube metadata into DB2. The IBM OLAP Center provides wizards to help you import the file. For more information, see the IBM documentation.

To import the XML file, use the following guidelines:

- Using the IBM OLAP Center tool, connect to the DB2 database.
- In the Import Wizard, choose the XML file that you want to import.
- If metadata exists that refers to database constructs that are not in the database, then an error message is displayed.
- When the wizard asks for an import option, choose to replace existing objects.
- When you are returned to the IBM OLAP Center, a diagram of the cube model is shown.

B.4.2.2.2 Guidelines for Changing Cube Metadata After Importing the XML File After you import the XML file, you might need to perform the following actions:

- Because Oracle OLAP does not store foreign keys as metadata, they will not exist in the converted metadata in the DB2 database. You must use the IBM Referential Integrity Utility for IBM Cube Views to generate foreign key informational constraints. You can obtain this utility on the IBM Web site.
- You might encounter other issues such as foreign key join columns being nullable. You can use the following methods to solve this problem:
  - If data in these columns are not null, then you should convert these columns to not-null columns.
  - If data in these columns are null or you prefer not to convert the column data type even if the column data is not null, then you should modify the cube model using the following guidelines:
    - In a fact-to-dimension join, you must manually eliminate this dimension object from the converted cube model and create a degenerated dimension object consisting of the foreign key of this join.
    - In a dimension-to-dimension join, you must manually eliminate the dimension object that represents the primary-key side of the join from the converted cube model and create a degenerated dimension object consisting of the foreign key of this join.
    - In a fact-to-fact join, you must manually eliminate the fact object that represents the primary-key side of the join from the converted cube model and create a degenerated dimension object consisting of the foreign key of this join.
- No incremental metadata changes will be allowed by the Cube Generator. Schema changes require that you manually delete cube model metadata in the DB2 database and convert the Oracle Business Intelligence metadata again. For example, if you must make a change to a dimension in a cube in the Oracle Business Intelligence metadata repository, then you must delete the cube model in the DB2 database, regenerate the XML file from the Oracle Business Intelligence repository, and import it into the DB2 database.
- You cannot delete metadata using the DB2 Cube Views Generator. Instead, you must manually delete the cube model using the IBM OLAP Center.
- The IBM Statistics tool and IBM Optimization Advisor must be run periodically.

For more information, see the IBM documentation.

B.4.2.3 Guidelines for Creating Materialized Query Tables (MQTs)

For more information, see the IBM documentation.

After you import the cube metadata into the database, you must run the IBM Optimization Advisor to generate SQL scripts and then execute those scripts to create the MQTs. You must provide certain parameters to the IBM Optimization Advisor to get optimal results from the implementation. The IBM Optimization Advisor wizard analyzes your metadata and recommends how to build summary tables that store and index aggregated data for SQL queries. Running the IBM Optimization Advisor can help you keep the MQTs current. Additionally, you must refresh your database after each ETL.

To create MQTs, use the following guidelines:

- In the IBM OLAP Center, choose the cube model that you want to optimize and open the IBM Optimization Advisor wizard.
- Follow the instructions in the wizard, using the following table as a guide.

When asked for:	Choose:
Summary Tables	Choose Deferred (or Immediate) and provide a tablespace for the tables
Limitations	Choose an appropriate value for the optimization parameters. You should turn on the Data-sampling option.
SQL Scripts	Creation of the scripts needed to run to create the Summary tables. Choose the filename and locations

- When the IBM Optimization Advisor closes, you must execute the SQL scripts to create the MQTs.

C

Managing Security for Dashboards and Analyses

This appendix explains how to manage security for dashboards and analyses such that users have only:

- Access to objects in the Oracle BI Presentation Catalog that are appropriate to them.
- Access to features and tasks that are appropriate to them.

For detailed information about securing an Oracle Business Intelligence system, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

This appendix contains the following sections:

- Section C.1, "Managing Security for Users of Oracle BI Presentation Services"
- Section C.2, "Managing Users Using Administration Pages"
- Section C.3, "Inheritance of Permissions and Privileges for Oracle BI Presentation Services"
- Section C.4, "Providing Shared Dashboards for Users"
- Section C.5, "Enabling Users to Act for Others"

C.1 Managing Security for Users of Oracle BI Presentation Services

System administrators must configure a business intelligence system to ensure that all functionality (including administrative functionality) is secured so that only authorized users can access the system to perform appropriate operations. Administrators also must be able to configure the system to secure all middle-tier communications.

This overview section contains the following topics:

- Section C.1.1, "Where Are Oracle BI Presentation Services Security Settings Made?"
- Section C.1.2, "What are the Security Goals in Oracle BI Presentation Services?"
- Section C.1.3, "How Are Permissions and Privileges Assigned to Users?"

C.1.1 Where Are Oracle BI Presentation Services Security Settings Made?

Security settings that affect users of Presentation Services are made in the following Oracle Business Intelligence components:

- **Oracle BI Administration Tool** — Enables you to perform the following tasks:

  - Set permissions for business models, tables, columns, and subject areas.
  - Specify database access for each user.
  - Specify filters to limit the data accessible by users.
  - Set authentication options.

  For information, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

- **Oracle BI Presentation Services Administration** — Enables you to set privileges for users to access features and functions such as editing views and creating agents and prompts.
- **Oracle BI Presentation Services** — Enables you to assign permissions for objects in the Oracle BI Presentation Catalog.

  In previous releases, you could assign permissions to objects from the Presentation Services Administration pages. In this release, you set permissions either in the Catalog Manager or the Catalog page of Presentation Services. See *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for information on assigning permissions in Presentation Services.

- **Catalog Manager** — Enables you to set permissions for Oracle BI Presentation Catalog objects. For information on Catalog Manager, see Chapter 11, "Configuring Connections to External Systems."

C.1.2 What are the Security Goals in Oracle BI Presentation Services?

When maintaining security in Presentation Services, you must ensure the following:

- Only the appropriate users can sign in and access Presentation Services. You must assign sign-in rights and authenticate users through the Oracle BI Server.

  Authentication is the process of using a user name and password to identify someone who is logging on. Authenticated users are then given appropriate authorization to access a system, in this case Presentation Services. Presentation Services does not have its own authentication system; it relies on the authentication system that it inherits from the Oracle BI Server.

  All users who sign in to Presentation Services are granted the AuthenticatedUser role and any other roles that they were assigned in Fusion Middleware Control.

  For information about authentication, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

- Users can access only the objects that are appropriate to them. You apply access control in the form of permissions, as described in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.
- Users have the ability to access features and functions that are appropriate to them. You apply user rights in the form of privileges. Example privileges are "Edit systemwide column formats" and "Create agents."

  Users are either granted or denied a specific privilege. These associations are created in a privilege assignment table, as described in Section C.2.3, "Managing Presentation Services Privileges."

You can configure Oracle Business Intelligence to use the single sign-on feature from the Web server. Presentation Services can use this feature when obtaining information for end users. For complete information on single sign-on, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

C.1.3 How Are Permissions and Privileges Assigned to Users?

When you assign permissions and privileges in Presentation Services, you can assign them in one of the following ways:

- To application roles — This is the recommended way of assigning permissions and privileges. Application roles provide much easier maintenance of users and their assignments. An application role defines a set of permissions granted to a user or group that has that role in the system's identity store. An application role is assigned in accordance with specific conditions. As such, application roles are granted dynamically based on the conditions present at the time authentication occurs.

  See *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for information on application roles.

- To individual users — You can assign permissions and privileges to specific users, but such assignments can be more difficult to maintain and so this approach is not recommended.

- To Catalog groups — This approach is maintained for backward compatibility with previous releases only.

  See Section C.2.2, "Working with Catalog Groups" for information on Catalog groups.

C.2 Managing Users Using Administration Pages

You can use the Administration pages in Oracle BI Presentation Services to perform the tasks that are described in the following sections:

- Section C.2.1, "Understanding the Administration Pages"
- Section C.2.2, "Working with Catalog Groups"
- Section C.2.3, "Managing Presentation Services Privileges"
- Section C.2.4, "Managing Sessions in Oracle BI Presentation Services"

C.2.1 Understanding the Administration Pages

The main Administration page contains links that allow you to display other administration pages for performing various functions, including those related to users in Presentation Services. You can obtain information about all these pages by clicking the Help button in the upper-right corner.

Note: Use care if multiple users have access to the Administration pages, because they can overwrite each other's changes. Suppose UserA and UserB are both accessing and modifying the Manage Privileges page. If UserA saves updates to privileges while UserB is also editing them, then UserB's changes are overwritten by those that UserA saved.

C.2.2 Working with Catalog Groups

In previous releases, Presentation Services groups were used for organizing users. Presentation Services group membership was used to determine the permissions and privileges that are associated with a user, either by explicit assignment or inheritance. In this release, Presentation Services groups have the following characteristics:

- Are referred to as Catalog groups.
- Can contain users, application roles, or other Catalog groups.
- Exist only for the purposes of compatibility with previous releases and only with Presentation Services.
- No longer have their own passwords.

While you can continue to use Catalog groups, it is recommended that you move to the use of application roles rather than Catalog groups for organizing users.

Presentation Services administrators must ensure that the names of Catalog groups are different from any user IDs that are used to log in to Oracle BI Presentation Services. If a user and a Catalog group share the same name, then the user receives an Invalid Account message when attempting to log in to Oracle BI Presentation Services.

On the Administration page in Presentation Services, you can perform the tasks that are described in the following sections:

- Section C.2.2.1, "Creating Catalog Groups"
- Section C.2.2.2, "Deleting Catalog Groups"
- Section C.2.2.3, "Editing Catalog Groups"

C.2.2.1 Creating Catalog Groups

To create Catalog groups:

1. From the Home page in Presentation Services, select **Administration**.
2. Click the **Manage Catalog Groups** link.
3. Click **Create a New Catalog Group**.
4. In the Add Group dialog, enter a name for the group.
5. Use the shuttle control to select the Catalog groups, users, and application roles to include in this group.

   > **Tip:** It is best practice to not include application roles in Catalog groups, to avoid complex group inheritance and maintenance situations. In particular do not add the AuthenticatedUser role to any other Catalog groups that you create. This ensures that only the desired Catalog groups (and users) have the specified permissions and privileges, by preventing users or authenticated users from unintentionally inheriting permissions and privileges from another Catalog group.

6. Click **OK**.

C.2.2.2 Deleting Catalog Groups

To delete Catalog groups:

1. From the Home page in Presentation Services, select **Administration**.
2. Click the **Manage Catalog Groups** link.
3. On the Manage Catalog Groups page, select the one or more groups to delete.

To help you locate the group that you want, enter text in the **Name** field and click **Search**.

4. Click **Delete Selected Groups**.
5. Click **OK** to confirm the deletion.

C.2.2.3 Editing Catalog Groups

To edit Catalog groups:

1. From the Home page in Presentation Services, select **Administration**.
2. Click the **Manage Catalog Groups** link.
3. On the Manage Catalog Groups page, select the group to edit.

   To help you locate the group that you want, enter text in the **Name** field and click **Search**.

   You can click the **More Groups** button to display the next 25 groups in the list.
4. In the Edit Group dialog, change the name or add or remove application roles, Catalog groups, and users.
5. Click **OK**.

C.2.3 Managing Presentation Services Privileges

This section contains the following topics about privileges in Presentation Services:

- Section C.2.3.1, "What are Privileges?"
- Section C.2.3.2, "Setting Privileges in Oracle BI Presentation Services Administration."
- Section C.2.3.3, "Default Oracle BI Presentation Services Privilege Assignments."

C.2.3.1 What are Privileges?

Privileges control the rights that users have to access the features and functionality of Oracle BI Presentation Services. Privileges are granted or denied to specific application roles, individual users, and Catalog groups using a privilege assignment table.

Like permissions, privileges are either explicitly set or are inherited through role or group membership. Explicitly denying a privilege takes precedence over any granted, inherited privilege. For example, if a user is explicitly denied access to the privilege to edit column formulas, but is a member of an application role that has inherited the privilege, then the user cannot edit column formulas.

Privileges are most commonly granted to the BIAuthor or BIConsumer roles. This allows users access to common features and functions of Presentation Services. While you can continue to grant privileges to Catalog groups, it is recommended that you switch the grants to application roles.

C.2.3.2 Setting Privileges in Oracle BI Presentation Services Administration

On the Manage Privileges Administration page in Presentation Services, you can view and administer privileges for application roles, individual users, and Catalog groups.

To administer privileges:

1. From the Home page in Presentation Services, select **Administration**.
2. Click the **Manage Privileges** link.

3. Click the link associated with the privilege to administer.
4. In the Privileges dialog, perform the following tasks:
   - To change the setting for an application role, individual user, or Catalog group that is listed in the dialog, select either **Granted** or **Denied** in the Permission column.
   - To add application roles, individual users, or Catalog groups for the privilege, click the **Add Users/Roles** button and complete the Add Application Roles, Catalog Groups, and Users dialog.
5. Click **OK**.
6. Click **Back**.

C.2.3.3 Default Oracle BI Presentation Services Privilege Assignments

Table C–1 lists the privileges that you can manage, along with the application role that is granted access to that privilege by default.

These privileges apply to the Oracle Business Intelligence infrastructure. If your organization uses prebuilt applications, then some privileges might be pre configured. For more information, see the documentation for the application.

Table C–1 Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
Access	Access to Dashboards	Allows users to view dashboards.	BIConsumer
Access	Access to Answers	Allows users to access the basic features of the Analysis editor.	BIAuthor
Access	Access to Delivers	Allows users to create and edit agents.	BIAuthor
Access	Access to Briefing Books	Allows users to view and download briefing books.	BIConsumer
Access	Access to Administration	Allows users to access the Administration pages in Presentation Services,	BIAdministrator
Access	Access to Segments	Allows users to access segments in Oracle's Siebel Marketing.	BIConsumer
Access	Access to Segment Trees	Allows users to access segment trees in Oracle's Siebel Marketing.	BIAuthor
Access	Access to List Formats	Allows users to access list formats in Oracle's Siebel Marketing.	BIAuthor
Access	Access to Metadata Dictionary	Allows users to access the metadata dictionary information for subject areas, folders, columns, and levels. For more information, see Section 19.2.1, "Providing Access to Metadata Dictionary Information."	BIAdministrator
Access	Access to Oracle BI for Microsoft Office	See Section C.2.3.3.2, "Access to Oracle BI for Microsoft Office Privilege."	BIConsumer
Access	Access to Conditions	Allows users to create conditions.	BIAuthor
Access	Access to KPI Builder	Allows users to create KPIs.	BIAuthor
Access	Access to Scorecard	Allows users access to Oracle BI Scorecard.	BIConsumer

Table C–1 (Cont.) Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
Actions	Create Navigate Actions	See Section C.2.3.3.1, "Access to Oracle BI Enterprise Edition Actions."	BIAuthor
Actions	Create Invoke Actions	See Section C.2.3.3.1, "Access to Oracle BI Enterprise Edition Actions."	BIAuthor
Actions	Save Actions Containing Embedded HTML	See Section C.2.3.3.1, "Access to Oracle BI Enterprise Edition Actions."	BIAdministrator
Admin: Catalog	Change Permissions	Allows users to modify permissions for catalog objects.	BIAuthor
Admin: Catalog	Toggle Maintenance Mode	Shows the Toggle Maintenance Mode link on the Presentation Services Administration page, which allows users to turn maintenance mode on and off. In maintenance mode, the catalog is read-only; no one can write to it.	BIAdministrator
Admin: General	Manage Sessions	Shows the Manage Sessions link on the Presentation Services Administration page, which displays the Manage Sessions page in which users manage sessions.	BIAdministrator
Admin: General	Manage Dashboards	Allows users to create and edit dashboards, including editing their properties.	BIAdministrator
Admin: General	See Session IDs	Allows users to see session IDs on the Manage Sessions page.	BIAdministrator
Admin: General	Issue SQL Directly	Shows the Issue SQL link on the Presentation Services Administration page, which displays the Issue SQL page in which users enter SQL statements.	BIAdministrator
Admin: General	View System Information	Allows users to view information about the system at the top of the Administration page in Presentation Services.	BIAdministrator
Admin: General	Performance Monitor	Allows users to monitor performance.	BIAdministrator
Admin: General	Manage Agent Sessions	Shows the Manage Agent Sessions link on the Presentation Services Administration page, which displays the Manage Agent Sessions page in which users manage agent sessions.	BIAdministrator
Admin: General	Manage Device Types	Shows the Manage Device Types link on the Presentation Services Administration page, which displays the Manage Device Types page in which users manage device types for agents.	BIAdministrator
Admin: General	Manage Map Data	Shows the Manage Map Data link on the Presentation Services Administration page, which displays the Manage Map Data page in which users edit layers, background maps, and images for map views.	BIAdministrator
Admin: General	See Privileged Errors	Allows users to see privileged error messages. Users can see detailed error messages about database connections or other details when lower level components fail.	BIAdministrator

Table C–1 (Cont.) Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
Admin: General	See SQL Issued in Errors	Allows users to see SQL statements that are returned by the BI Server in error messages.	BIConsumer
Admin: General	Manage Marketing Jobs	Shows the Manage Marketing Jobs link on the Presentation Services Administration page, which displays the Marketing Job Management page in which users manage marketing jobs.	BIAuthor
Admin: General	Manage Marketing Defaults	Shows the Manage Marketing Defaults link on the Presentation Services Administration page, which displays the Manage Marketing Defaults page in which users manage defaults for Oracle's Siebel Marketing application.	BIAdministrator
Admin: Security	Manage Catalog Groups	Shows the Manage Catalog Groups link on the Presentation Services Administration page, which displays the Manage Catalog Groups page in which users edit Catalog groups.	BIAdministrator
Admin: Security	Manage Privileges	Shows the Manage Privileges link on the Presentation Services Administration page, which displays the Manage Privileges page in which users manage the privileges that are described in this table.	BIAdministrator
Admin: Security	Set Ownership of Catalog Objects	Allows users to edit the ownership of objects in the catalog on the Catalog page.	BIAdministrator
Admin: Security	User Population - Can List Users	Allows users to see the list of users for which they can perform tasks such as assigning privileges and permissions.	BIConsumer, BISystem
Admin: Security	User Population - Can List Groups	Allows users to see the list of groups for which they can perform tasks such as assigning privileges and permissions.	BIConsumer, BISystem
Briefing Book	Add To or Edit a Briefing Book	Allows users to see the Add to Briefing Book link on dashboard pages and analyses and the Edit link in briefing books.	BIAuthor
Briefing Book	Download Briefing Book	Allows users to download briefing books.	BIConsumer
Catalog	Personal Storage	Allows users to have write access to their own My Folders folders and can create content there. If users do not have this privilege, then they can receive email alerts but cannot receive dashboard alerts.	BIConsumer
Catalog	Reload Metadata	Allows users to click the **Reload Server Metadata** link from the Refresh menu in the toolbar of the Subject Areas pane.	BIAdministrator
Catalog	See Hidden Items	Allows users to see hidden items in catalog folders. Users can also select the **Show Hidden Items** box on the Catalog page.	BIAuthor
Catalog	Create Folders	Allows users to create folders in the catalog.	BIAuthor
Catalog	Archive Catalog	Allows users to archive the folders and objects in the catalog.	BIAdministrator

Table C–1 (Cont.) Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
Catalog	Unarchive Catalog	Allows users to unarchive catalog objects that have been archived previously.	BIAdministrator
Catalog	Upload Files	Allows users to upload files into an existing catalog.	BIAdministrator
Conditions	Create Conditions	Allows users to create or edit named conditions.	BIAuthor
Dashboards	Save Customizations	See Section 19.5, "Controlling Access to Saved Customization Options in Dashboards."	BIConsumer
Dashboards	Assign Default Customizations	See Section 19.5, "Controlling Access to Saved Customization Options in Dashboards."	BIAuthor
Formatting	Save SystemWide Column Formats	Allows users to save systemwide defaults when specifying formats for columns.	BIAdministrator
My Account	Access to My Account	Allows users to access the My Account dialog.	BIConsumer
My Account	Change Preferences	Allows users to access the Preferences tab of the My Account dialog.	BIConsumer
My Account	Change Delivery Options	Allows users to access the Delivery Options tab of the My Account dialog.	BIConsumer
Answers	Create Views	Allows users to create views.	BIAuthor
Answers	Create Prompts	Allows users to create prompts.	BIAuthor
Answers	Access Advanced Tab	Allows users to access the Advanced tab in the Analysis editor.	BIAuthor
Answers	Edit Column Formulas	Allows users to edit column formulas.	BIAuthor
Answers	Save Content with HTML Markup	Allows users to save objects such as views and actions that contain HTML code.	BIAdministrator
Answers	Enter XML and Logical SQL	Allows users to use the Advanced SQL tab.	BIAuthor
Answers	Edit Direct Database Analysis	Allows users to create and edit requests that are sent directly to the back-end data source.	BIAdministrator
Answers	Create Analysis from Simple SQL	Allows users to select the **Create Analysis from Simple SQL** option in the Select Subject Area list.	BIAdministrator
Answers	Create Advanced Filters and Set Operations	Allows users to click the **Combine results based on union, intersection, and difference operations** button from the Criteria tab in the Analysis editor.	BIAuthor
Answers	Save Filters	Allows users to save filters	BIAuthor
Answers	Execute Direct Database Analysis	Allows users to issue requests directly to the back-end data source.	BIAdministrator
Delivers	Create Agents	Allows users to create agents.	BIAuthor
Delivers	Publish Agents for Subscription	Allows users to publish agents for subscription.	BIAuthor

Table C–1 (Cont.) Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
Delivers	Deliver Agents to Specific or Dynamically Determined Users	Allows users to deliver agents to other users.	BIAdministrator
Delivers	Chain Agents	Allows users to chain agents.	BIAuthor
Delivers	Modify Current Subscriptions for Agents	Allows users to modify the current subscriptions for agents, including unsubscribing users.	BIAdministrator
Proxy	Act As Proxy	Allows users to act as proxy users for other users, as described in Section C.5, "Enabling Users to Act for Others."	Denied: BIConsumer
RSS Feeds	Access to RSS Feeds	Allows users to subscribe to and receive RSS feeds with alerts and contents of folders. If Presentation Services uses the HTTPS protocol, then the RSS Reader that you use must also support the HTTPS protocol.	BIAuthor
Scorecard	Create/Edit Scorecards	Allows users to create and edit scorecards.	BIAuthor
Scorecard	View Scorecards	Allows users to view scorecards.	BIConsumer
Scorecard	Create/Edit Objectives	Allows users to create and edit objectives.	BIAuthor
Scorecard	Create/Edit Initiatives	Allows users to create and edit initiatives.	BIAuthor
Scorecard	Create Views	Allows users to create and edit scorecard views, such as strategy trees.	BIAuthor
Scorecard	Create/Edit Causes and Effects Linkages	Allows users to create and edit cause and effect relationships.	BIAuthor
Scorecard	Create/Edit Perspectives	Allows users to create and edit perspectives.	BIAdministrator
Scorecard	Add Annotations	Allows users to add comments to KPIs and scorecard components.	BIConsumer
Scorecard	Override Status	Allows users to override statuses of KPIs and scorecard components.	BIConsumer
Scorecard	Create/Edit KPIs	Allows users to create and edit KPIs.	BIAuthor
Scorecard	Add Scorecard Views to Dashboards	Allows users to add scorecard views (such as strategy trees) to dashboards.	BIConsumer
List Formats	Create List Formats	Allows users to create list formats in Oracle's Siebel Marketing.	BIAuthor
List Formats	Create Headers and Footers	Allows users to create headers and footers for list formats in Oracle's Siebel Marketing.	BIAuthor
List Formats	Access Options Tab	Allows users to access the Options tab for list formats in Oracle's Siebel Marketing.	BIAuthor
List Formats	Add/Remove List Format Columns	Allows users to add and remove columns for list formats in Oracle's Siebel Marketing.	BIAdministrator
Segmentation	Create Segments	Allows users to create segments in Oracle's Siebel Marketing.	BIAuthor
Segmentation	Create Segment Trees	Allows users to create segment trees in Oracle's Siebel Marketing.	BIAuthor
Segmentation	Create/Purge Saved Result Sets	Allows users to create and purge saved result sets in Oracle's Siebel Marketing.	BIAdministrator

Table C–1 (Cont.) Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
Segmentation	Access Segment Advanced Options Tab	Allows users to access the Segment Advanced Options tab in Oracle's Siebel Marketing.	BIAdministrator
Segmentation	Access Segment Tree Advanced Options Tab	Allows users to access the Segment Tree Advanced Options tab in Oracle's Siebel Marketing.	BIAdministrator
Segmentation	Change Target Levels within Segment Designer	Allows users to change target levels within the Segment Designer in Oracle's Siebel Marketing.	BIAdministrator
SOAP	Access SOAP	Allows users to access various web services.	BIConsumer, BISystem
SOAP	Impersonate as System User	Allows users to impersonate a system user using a web service.	BISystem
SOAP	Access MetadataService Service	Allows users to access the MetadataService web service.	BIConsumer, BISystem
SOAP	Access AnalysisExportViewsService Service	Allows users to access the ReportingEditingService web service.	BIConsumer
SOAP	Access ReportingEditingService Service	Allows users to access the ReportingEditingService web service.	BIConsumer, BISystem
SOAP	Access ConditionEvaluationService Service	Allows users to access the ConditionEvaluationService web service.	BIConsumer, BISystem
SOAP	Access ReplicationService Service	Allows users to access the ReplicationService web service to replicate the Oracle BI Presentation Catalog.	BISystem
SOAP	Access CatalogIndexingService Service	Allows users to access the CatalogIndexingService web service to index the Oracle BI Presentation Catalog for use with full-text search.	BISystem
SOAP	Access DashboardService Service	Allows users to access the DashboardService web service.	BIConsumer, BISystem
SOAP	Access SecurityService Service	Allows users to access the SecurityService web service.	BIConsumer, BISystem
SOAP	Access ScorecardMetadataService Service	Allows users to access the ScorecardMetadataService web service.	BIConsumer, BISystem
SOAP	Access ScorecardAssessmentService Service	Allows users to access the ScorecardAssessmentService web service.	BIConsumer, BISystem
SOAP	Access HtmlViewService Service	Allows users to access the HtmlViewServiceService web service.	BIConsumer, BISystem
SOAP	Access CatalogService Service	Allows users to access the CatalogService web service.	BIConsumer, BISystem
SOAP	Access IBotService Service	Allows users to access the IBotService web service.	BIConsumer, BISystem

Table C–1 (Cont.) Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
SOAP	Access XmlGenerationService Service	Allows users to access the XmlGenerationService web service.	BIConsumer, BISystem
SOAP	Access JobManagementService Service	Allows users to access the JobManagementService web service.	BIConsumer, BISystem
SOAP	Access KPIAssessmentService Service	Allows users to access the JKPIAssessmentService web service.	BIConsumer, BISystem
Subject Area (*by its name*)	Access within Oracle BI Answers	Allows users to access the specified subject area within the Answers editor.	BIAuthor
View Analyzer	Add/Edit AnalyzerView	Allows users to access the Analyzer view.	BIAdministrator
View Column Selector	Add/Edit Column SelectorView	Allows users to create and edit column selector views.	BIAuthor
View Compound	Add/Edit CompoundView	Allows users to create and edit compound layouts.	BIAuthor
View Graph	Add/Edit GraphView	Allows users to create and edit graph views.	BIAdministrator
View Funnel	Add/Edit FunnelView	Allows users to create and edit funnel graph views.	BIAuthor
View Gauge	Add/Edit GaugeView	Allows users to create and edit gauge views.	BIAuthor
View Filters	Add/Edit FiltersView	Allows users to create and edit filters.	BIAuthor
View Dashboard Prompt	Add/Edit Dashboard PromptView	Allows users to create and edit dashboard prompts.	BIAuthor
View Static Text	Add/Edit Static TextView	Allows users to create and edit static text views.	BIAuthor
View Legend	Add/Edit Legend View	Allows users to create and edit legend views.	BIAuthor
View Map	Add/Edit MapView	Allows users to create and edit map views.	BIAuthor
View Narrative	Add/Edit NarrativeView	Allows users to create and edit narrative views.	BIAuthor
View Nested Request	Add/Edit Nested RequestView	Allows users to create and edit nested analyses.	BIAuthor
View No Results	Add/Edit No ResultsView	Allows users to create and edit no result views.	BIAuthor
View Pivot Table	Add/Edit Pivot TableView	Allows users to create and edit pivot table views.	BIAuthor
View Report Prompt	Add/Edit Report PromptView	Allows users to create and edit prompts.	BIAuthor
View Create Segment	Add/Edit Create SegmentView	Allows users to create and edit segment views.	BIAuthor
View Logical SQL	Add/Edit Logical SQLView	Allows users to create and edit logical SQL views.	BIAuthor
View Table	Add/Edit TableView	Allows users to create and edit table views.	BIAuthor

Table C–1 (Cont.) Privileges and Default Settings for the Oracle Business Intelligence Infrastructure

Component	Privilege	Description	Default Role Granted
View Create Target List	Add/Edit Create Target ListView	Allows users to create and edit target list views.	BIAuthor
View Ticker	Add/Edit TickerView	Allows users to create and edit ticker views.	BIAuthor
View Title	Add/Edit TitleView	Allows users to create and edit title views.	BIAuthor
View View Selector	Add/Edit View SelectorView	Allows users to create and edit view selector views.	BIAuthor
Write Back	Write Back to Database	Grants the right to write data into the data source.	Denied: BIConsumer
Write Back	Manage Write Back	Grants the right to manage write back requests.	BIAdministrator

C.2.3.3.1 Access to Oracle BI Enterprise Edition Actions You must set the Action privileges, which determine whether the Actions functionality is available to users and specify which user types can create Actions. The following list describes these privileges:

- **Create Navigate Actions —** Determines which users can create a Navigate action type. The sessions of users who are denied this privilege do not contain any of the user interface components that allow them to create Navigate Actions. For example, if a user is denied this privilege and chooses to create an action from the Oracle BI Enterprise Edition global header, the dialog where the user selects an action type does not include the Navigate Actions options (Go to BI Content, Go to a Web Page, and so on). However, users who are denied this privilege can add saved actions to analyses and dashboards. And, users who are denied this privilege can execute an action from an analysis or dashboard that contains an action.
- **Create Invoke Actions —** Determines which users can create an Invoke action type. The sessions of user who are denied this privilege do not contain any of the user interface components that allow them to create Invoke Actions. For example, if a user is denied this privilege and chooses to access the agent editor's Actions tab and clicks the **Add New Action** button, the dialog where the user selects the action type does not include the Invoke Actions options (Invoke a Web Service, Invoke an HTTP Request, and so on). However, users who are denied this privilege can add saved actions to analyses and dashboards. And, users who are denied this privilege can execute an action from an analysis or dashboard that contains an action.
- **Save Actions Containing Embedded HTML —** Determines which users can embed HTML code in the customization of web service action results. Use care in assigning this privilege, because it poses a security risk to allow users to run HTML code.

C.2.3.3.2 Access to Oracle BI for Microsoft Office Privilege The Access to Oracle BI for Microsoft Office privilege shows the following options for the **Download BI Desktop Tools** link in the Get Started area of the Oracle BI EE Home page:

- **Oracle BI for MS Office**: Downloads the installation file for the Oracle BI Add-in for Microsoft Office.
- **Smart View**: Downloads the installation file for Oracle Hyperion Smart View.

The Access to Oracle BI for Microsoft Office privilege does not affect the display of the **Copy** link for analyses. The link is always available there.

The location of the installation file to download for the Oracle BI Add-in for Microsoft Office is specified by default in the BIforOfficeURL element in the instanceconfig.xml file. For more information on using the Oracle BI Add-in for Microsoft Office and the **Copy** option, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

C.2.4 Managing Sessions in Oracle BI Presentation Services

Using the Session Management page in Presentation Services Administration, you can view information about active users and running analyses, cancel requests, and clear the cache.

To manage sessions in Presentation Services:

1. From the Home page in Presentation Services, select **Administration**.
2. Click the **Manage Sessions** link.

   The Session Management screen is displayed with the following tables:

   – The Sessions table, which gives information about sessions that have been created for users who have logged in:

   – The Cursor Cache table, which shows the status of analyses:

To cancel all running requests:

1. Click **Cancel Running Requests**.
2. Click **Finished**.

To cancel one running analysis:

- In the Cursor Cache table, identify the analysis and click the **Cancel** link in the **Action** column.

  The user receives a message indicating that the analysis was canceled by an administrator.

To clear the Web cache:

1. In the Cursor Cache table, identify the analysis and click **Close All Cursors**.
2. Click **Finished**.

To clear the cache entry associated with an analysis:

- In the Cursor Cache table, identify the analysis and click the **Close** link in the **Action** column.

To view the query file for information about an analysis:

- In the Cursor Cache table, identify the analysis and click the **View Log** link.

> **Note:** Query logging must be turned on for data to be saved in this log file.

C.3 Inheritance of Permissions and Privileges for Oracle BI Presentation Services

Permissions and privileges can be assigned to users directly or through membership in application roles or Catalog groups. From another perspective, permissions and privileges can be assigned explicitly or effectively. Effective permissions and privileges are assigned indirectly through inheritance from application roles or Catalog groups, which is the recommended approach for assignments.

This section contains the following topics:

- Section C.3.1, "Rules for Inheritance for Permissions and Privileges"
- Section C.3.2, "Example of Inherited Privileges for Application Roles"
- Section C.3.3, "Example of Inherited Privileges for Catalog Groups"

C.3.1 Rules for Inheritance for Permissions and Privileges

The following list describes the rules of inheritance for permissions and privileges:

- Any permissions or privileges granted explicitly to a user override any permissions or privileges inherited from the application roles or Catalog groups to which the user belongs.
- If a user belongs to two application roles or Catalog groups and both are granted permissions, then the least restrictive permissions are given to the user.

  For example, if one application role allows Open access and another allows Modify access, then the least restrictive access would be granted; in this example, Open access.

  > **Note:** The exception to this is if one of the two application roles or Catalog groups is explicitly denied the permissions, in which case the user is denied.

- If a user belongs to Application Role X, and Application Role X is a member of Application Role Y, then any permissions assigned to Application Role X override any permissions assigned to Application Role Y. The same holds true if X and Y are Catalog groups.

  For example, if Marketing has Open permissions, Marketing Administrators, which is a member of Marketing, can have Full Control permission.

- If a Catalog group is specified along with an application role in the Permissions dialog in Presentation Services, then the Catalog group takes precedence.

  For example, suppose that for a certain object, the BIAdministrator role has Read-Only permission and the Admin Catalog Group has Full Control permission. If a user signs in who is a member of both the BIAdministrator role and the Admin Catalog Group, then he is granted full access to the object.

- Explicitly denying access takes precedence over any other permissions or privileges.

C.3.2 Example of Inherited Privileges for Application Roles

Figure C–1 shows an example of how privileges are inherited through application roles. At the top of the diagram is a rectangle labeled User1, which specifies that User1 is a member of Role1 and Role2. Attached beneath the User1 rectangle are two more rectangles — one on the left that represents Role1 and one on the right that represents Role2.

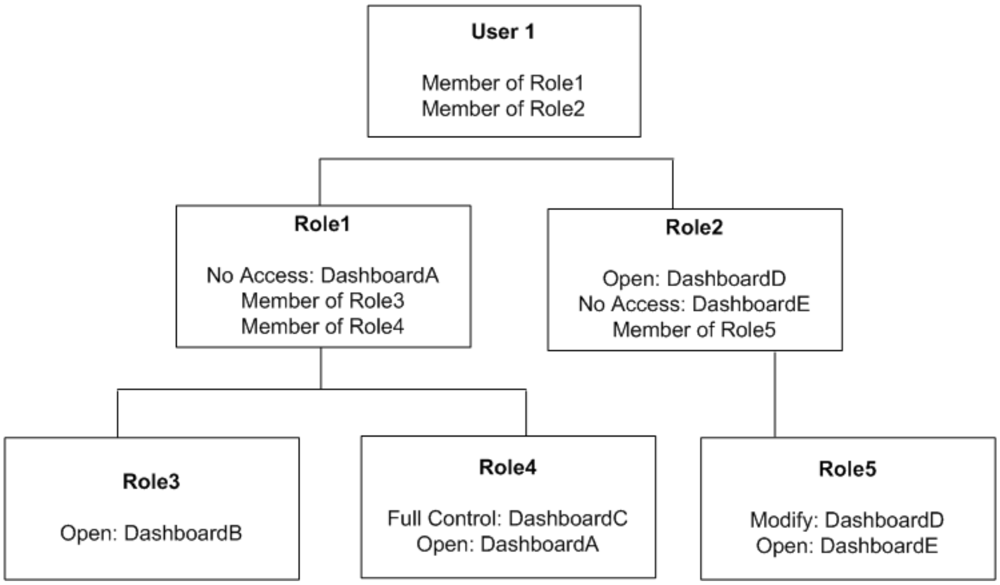
- The Role1 rectangle specifies that Role1 has no access to DashboardA and is a member of Role3 and Role4.
- The Role2 rectangle specifies that Role2 has Open access to DashboardD, is a member of Role5, and has no access to DashboardE.

Attached beneath the Role1 rectangle are two more rectangles — one on the left that represents Role3 and one on the right that represents Role4:

- The Role3 rectangle specifies that Role3 has Open access to DashboardB.
- The Role4 rectangle specifies that Role4 has Full Access to DashboardC and Open access to DashboardA.

And finally, attached beneath the Role2 rectangle is a rectangle that represents Role5. The Role5 rectangle specifies that Role5 has Modify access to Dashboard D and Open access to DashboardE.

Figure C–1 Example of Inheritance of Permissions Using Roles



In this example:

- User1 is a direct member of Role1 and Role2, and is an indirect member of Role3, Role4, and Role5.
- The permissions and privileges from Role1 are no access to DashboardA, Open access to DashboardB, and Full Control for DashboardC.
- If one application role is a member of a second application role, then any permissions assigned to the first application role override any permissions assigned to the first role. Therefore, the inherited permissions and privileges from Role2 include Modify access to DashboardD from Role5.
- Specifically denying access always takes precedence over any other settings. Therefore, Role1's denial of access to DashboardA overrides Role4's Open access. The result is that Role1 has no access to DashboardA. Likewise, Role5 has no

access to DashboardE, because access to that dashboard is explicitly denied for Role2.

The total permissions and privileges granted to User1 are as follows:

- No access to DashboardA and DashboardE, because access is specifically denied.
- Open access to DashboardB.
- Full Control for DashboardC.
- Modify access to DashboardD.

C.3.3 Example of Inherited Privileges for Catalog Groups

Any permissions or privileges granted explicitly to a Catalog group take precedence over permissions or privileges granted to an application role. For example, suppose that have an application role called Marketing_US that has Full Access to the Marketing Dashboard. You want to restrict a small set of the users in the Marketing_US role to not have access to that dashboard. To do so, you create a Catalog group called Marketing_SanJose and add the appropriate users as members of that group. You then deny the Marketing_SanJose Catalog group access to the Marketing Dashboard. Even though those users belong to the Marketing_US role, they are denied access to the Marketing Dashboard.

C.4 Providing Shared Dashboards for Users

This section contains the following topics on providing shared dashboards for users:

- Section C.4.1, "Understanding the Catalog Structure for Shared Dashboards"
- Section C.4.2, "Creating Shared Dashboards"
- Section C.4.3, "Testing the Dashboards"
- Section C.4.4, "Releasing Dashboards to the User Community"

C.4.1 Understanding the Catalog Structure for Shared Dashboards

The Oracle BI Presentation Catalog has two main folders:

- **My Folders** — Contains the personal storage for individual users. Includes a Subject Area Contents folder where you save objects such as calculated items and groups.
- **Shared Folders** — Contains objects and folders that are shared across users. Dashboards that are shared across users are saved in a Dashboards subfolder under a common subfolder under the /Shared Folders folder

Note: If a user is given permission to an analysis in the Oracle BI Presentation Catalog that references a subject area to which the user does not have permission, then the Oracle BI Server still prevents the user from executing the analysis.

C.4.2 Creating Shared Dashboards

After setting up the Oracle BI Presentation Catalog structure and setting permissions, you can create shared dashboards and content for use by others.

One advantage to creating shared dashboards is that pages that you create in the shared dashboard are available for reuse. Users can create their own dashboards using the pages from your shared dashboards and any new pages that they create. You can add pages and content as described in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

If you plan to allow multiple users to modify a shared default dashboard, then consider putting these users into an application role. For example, suppose that you create an application role called Sales and create a default dashboard called SalesHome. Of the 40 users that have been assigned the Sales application role, suppose that there are three who must have the ability to create and modify content for the SalesHome dashboard. Create a SalesAdmin application role, with the same permissions as the primary Sales application role. Add the three users who are allowed to make changes to the SalesHome dashboard and content to this new SalesAdmin application role, and give this role the appropriate permissions in the Oracle BI Presentation Catalog. This allows those three users to create and modify content for the SalesHome dashboard. If a user no longer requires the ability to modify dashboard content, then you can change the user's role assignment to Sales. If an existing Sales role user must have the ability to create dashboard content, then the user's role assignment can be changed to SalesAdmin.

For more information about creating shared dashboards, see Section 19.1, "Managing Dashboards."

C.4.3 Testing the Dashboards

Before releasing dashboards and content to the user community, perform some tests.

To test the dashboard:

1. Verify that users with appropriate permissions can correctly access it and view the intended content.
2. Verify that users without appropriate permissions cannot access the dashboard.
3. Verify that styles and skins are displayed as expected, and that other visual elements are as expected.
4. Correct any problems you find and test again, repeating this process until you are satisfied with the results.

C.4.4 Releasing Dashboards to the User Community

After testing is complete, notify the user community that the dashboard is available, ensuring that you provide the relevant network address.

C.5 Enabling Users to Act for Others

This section contains the following topics on enabling users to act for others:

- Section C.5.1, "Why Enable Users to Act for Others?"
- Section C.5.2, "What are the Proxy Levels?"
- Section C.5.3, "Process of Enabling Users to Act for Others"

C.5.1 Why Enable Users to Act for Others?

You can enable one user to act for another user in Oracle BI Presentation Services. When a user (called the proxy user) acts as another (called the target user), the proxy user can access the objects in the catalog for which the target user has permission.

Enabling a user to act for another is useful, for example, when a manager wants to delegate some of his work to one of his direct reports or when IT support staff wants to troubleshoot problems with another user's objects.

See *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for information on how users enable others to act for them.

C.5.2 What are the Proxy Levels?

When you enable a user to be a proxy user, you also assign an authority level (called the proxy level). The proxy level determines the privileges and permissions granted to the proxy user when accessing the catalog objects of the target user. The following list describes the proxy levels:

- **Restricted** — Permissions are read-only to the objects to which the target user has access. Privileges are determined by the proxy user's account (not the target user's account).

  For example, suppose a proxy user *has not* been assigned the Access to Answers privilege, and the target user *has*. When the proxy user is acting as the target user, the target user *cannot* access Answers.

- **Full** — Permissions and privileges are inherited from the target user's account.

  For example, suppose a proxy user *has not* been assigned the Access to Answers privilege, and the target user *has*. When the proxy user is acting as the target user, the target user *can* access Answers.

When you have enabled a user to act as a proxy user, that user can display the **Act As** option in the global header of Presentation Services to select the target user to act as, provided the Act As Proxy privilege has been set.

> **Tip:** Before a proxy user can act as a target user, the target user must have signed into Presentation Services at least once and accessed a dashboard.

C.5.3 Process of Enabling Users to Act for Others

To enable users to act for others, perform the following tasks:

- Section C.5.3.1, "Defining the Association Between Proxy Users and Target Users"
- Section C.5.3.2, "Creating Session Variables for Proxy Functionality"
- Section C.5.3.3, "Modifying the Configuration File Settings for Proxy Functionality"
- Section C.5.3.4, "Creating a Custom Message Template for Proxy Functionality"
- Section C.5.3.5, "Assigning the Proxy Privilege"

C.5.3.1 Defining the Association Between Proxy Users and Target Users

You define the association between proxy users and target users in the database by identifying, for each proxy user/target user association, the following:

- ID of the proxy user

- ID of the target user
- Proxy level (either full or restricted)

For example, you might create a table called Proxies in the database that looks like this:

proxyId	targetId	proxyLevel
Ronald	Eduardo	full
Timothy	Tracy	restricted
Pavel	Natalie	full
William	Sonal	restricted
Maria	Imran	restricted

After you define the association between proxy users and target users, you must import the schema to the physical layer of the Oracle BI Server. For information on importing a schema, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

C.5.3.2 Creating Session Variables for Proxy Functionality

To authenticate proxy users, you must create the following two session variables along with their associated initialization blocks. For both variables, you must modify the sample SQL statement according to the schema of the database.

- **PROXY** — Use this variable to store the name of the proxy user.

  Use the initialization block named ProxyBlock and include code such as the following:

  ```
  select targetId
  from Proxies
  where 'VALUEOF(NQ_SESSION.RUNAS)'=targetId and ':USER'=proxyId
  ```

- **PROXYLEVEL** — Use this optional variable to store the proxy level, either Restricted or Full. If you do not create the PROXYLEVEL variable, then the Restricted level is assumed.

  Use the initialization block named ProxyLevel and include code such as the following:

  ```
  select proxyLevel
  from Proxies
  where 'VALUEOF(NQ_SESSION.RUNAS)'=targetId and ':USER'=proxyId
  ```

For more information on creating session variables, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

C.5.3.3 Modifying the Configuration File Settings for Proxy Functionality

Use various elements in the instanceconfig.xml file to configure the proxy functionality.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually configure for proxy functionality:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the elements that are described in the following list:
   - LogonParam: Serves as the parent element for the TemplateMessageName and MaxValues elements.
   - TemplateMessageName: Specifies the name of the custom message template in the Custom Messages folder that contains the SQL statement to perform tasks related to displaying proxy and target users. The default name is LogonParamSQLTemplate.

     The name that you specify in the TemplateMessageName element must match the name that you specify in the WebMessage element in the custom message file. For more information, see Section C.5.3.4, "Creating a Custom Message Template for Proxy Functionality."
   - MaxValues: Specifies the maximum number of target users to be listed in the **User** box in the Act As dialog box. If the number of target users for a proxy user exceeds this value, then an edit box, where the proxy user can type the ID of a target user, is shown rather than a list of target users. The default is 200.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<LogonParam>
     <TemplateMessageName>LogonParamSQLTemplate</TemplateMessageName>
     <MaxValues>100</MaxValues>
</LogonParam>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

C.5.3.4 Creating a Custom Message Template for Proxy Functionality

You must create a custom message template for the proxy functionality that contains the SQL statement to perform the following tasks:

- Obtain the list of target users that a proxy user can act as. This list is displayed in the User box in the Act As dialog box.
- Verify whether the proxy user can act as the target user.
- Obtain the list of proxy users that can act as the target user. This list is displayed on the target user's My Account screen.

In the custom message template, you place the SQL statement to retrieve this information in the following XML elements:

Element	Description
getValues	Specifies the SQL statement to return the list of target users and corresponding proxy levels. The SQL statement must return either one or two columns, where the: ■ First column returns the IDs of the target users ■ (Optional) Second column returns the names of the target users

Element	Description
verifyValue	Specifies the SQL statement to verify if the current user can act as the specified target user. The SQL statement must return at least one row if the target user is valid or an empty table if the target user is invalid.
getDelegateUsers	Specifies the SQL statement to obtain the list of proxy users that can act as the current user and their corresponding proxy levels. The SQL statement must return either one or two columns, where the: ■ First column returns the names of the proxy users ■ (Optional) Second column returns the corresponding proxy levels

For information on the directory for storing custom message templates, see Section 22.5.3, "Customizing XML Messages." You can create the custom message template in one of the following files:

- The original custom message file in the directory
- A separate XML file in the directory

To create the custom message template:

1. To create the custom message template in the original custom message file:

   a. Make a backup of the original custom message file in a separate directory.

   b. Make a development copy in a different directory and open it in a text or XML editor.

2. To create the custom message template in a separate XML file, create and open the file in the *ORACLE_INSTANCE*\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips*n*\msgdb\l_*xx*\customMessages directory.

3. Start the custom message template by adding the WebMessage element's begin and end tags. For example:

```
<WebMessage name="LogonParamSQLTemplate">
</WebMessage>
```

> **Note:** The name that you specify in the WebMessage element must match the name that you specify in the TemplateMessageName element in the instanceconfig.xml file. For information, see Section C.5.3.3, "Modifying the Configuration File Settings for Proxy Functionality."

4. After the </WebMessage> tag:

   a. Add the <XML> and </XML> tags

   b. Between the <XML> and </XML> tags, add the <logonParam name="RUNAS"> and </logonParam> tags.

   c. Between the <logonParam name="RUNAS"> and </logonParam> tags, add each of the following tags along with its corresponding SQL statements:

      * <getValues> and </getValues>

  * <verifyValue> and </verifyValue>
  * <getDelegateUsers> and </getDelegateUsers>

The following entry is an example:

```
<XML>
     <logonParam name="RUNAS">
     <getValues>select targetId
                from Proxies
                where proxyId='@{USERID}'</getValues>
     <verifyValue>select targetId
                from Proxies
                  where proxyId ='@{USERID}'
             and targetId='@{VALUE}'</verifyValue>
     <getDelegateUsers>select proxyId, proxyLevel
                from Proxies
                  where targetId ='@{USERID}'
     </getDelegateUsers>
     </logonParam>
</XML>
```

(Note that you must modify the example SQL statement according to the schema of the database.)

5. If you created the custom message template in the development copy of the original file, then replace the original file in the customMessages directory with the newly edited file.
6. Test the new file.
7. (Optional) If you created the custom message template in the development copy of the original file, then delete the backup and development copies.
8. Load the custom message template by either restarting the server or by clicking the **Reload Files and Metadata** link on the Presentation Services Administration screen. For information on the Administration page, see Section C.2.1, "Understanding the Administration Pages."

C.5.3.5 Assigning the Proxy Privilege

For each user whom you want to enable as a proxy user or for each application role or Catalog group whose members you want to enable as proxy users, you must grant the Act As Proxy privilege. For information on how to assign privileges, see Section C.2.3.2, "Setting Privileges in Oracle BI Presentation Services Administration."

D

Setting Up the SA System Subject Area

Oracle BI Presentation Services delivers alerts from Oracle BI Delivers to specified e-mail addresses, phone numbers, and so on. These delivery destinations are stored in the Oracle Business Intelligence profile for each user. In some cases, you might want to automatically populate the phone numbers or e-mail addresses in user profiles.

If you must automate only the population of e-mail addresses in user profiles, then you should populate the e-mail address field for users in your LDAP server or other authentication provider, if possible. These values are used to populate the e-mail address in Oracle Business Intelligence user profiles, enabling users to receive content from Delivers, even if they have not signed into Oracle Business Intelligence. This feature works for any LDAP server that has a mail attribute for its users.

In some cases, however, you might want to automatically populate additional user profile options, in addition to e-mail addresses. For example, you might want to automatically populate a cell phone number as part of the user profile information, if you want Delivers to deliver a format suitable for a cell phone (like text) using an e-mail gateway.

In this situation, you can configure a special subject area in the repository called SA System that retrieves user information from a database and populates the user profile information. This appendix explains how to configure and use the SA System subject area to accomplish this task.

If you choose to use the SA System subject area, then you should discourage users from configuring delivery profiles on their own. By default, values that are specified in delivery profiles take precedence over values that are shown in the SA System subject area.

This appendix contains the following topics:

- Section D.1, "About the SA System Subject Area"
- Section D.2, "Setting Up the Data Source for the SA System Subject Area"
- Section D.3, "Importing SA System Data Into the Repository"
- Section D.4, "Setting Configuration Options for the SA System Subject Area"

D.1 About the SA System Subject Area

In previous releases of Oracle Business Intelligence, **SA System** was a subject area that exposed group membership to Delivers and enabled contact information, such as e-mail addresses to be retrieved from a database and used as delivery devices in Delivers. The SA System subject area feature automatically populated delivery devices and profiles for users instead of requiring users to update their My Account screen in

Delivers. The SA System subject area provided the users associated with each group and external e-mail addresses to Delivers.

In this release of Oracle Business Intelligence, Delivers still must determine group and role membership so that it can appropriately deliver alerts. Typically, however, your LDAP identity store is now the source of group and role membership. If SA System is defined and enabled, then membership of application roles and catalog groups is derived from the SA System subject area in Delivers. The names of the applications roles and catalog groups that are selected in an agent are used to determine group membership in the SA System subject area.

Note that you do not need SA System if you are using an LDAP server and you must populate only user profile e-mail addresses. The recommended best practice for populating e-mail addresses in user profiles is to use the mail attribute in your LDAP server. Because most portable devices can read e-mail directly, specific text or SMS formats are often not required for agent delivery, and populating e-mail addresses from LDAP is usually sufficient.

Also note that you do not need to use SA system to use the feature **Get Recipients from the Analysis Used in the Agent Condition**. Instead, this feature is used when the recipients can be determined from the query results and the data to be delivered is specific to those users.

Note that it is possible to configure initialization block-based user authentication using the tables in SA System as a source for the user population. Using the SA System data in this way is separate from using SA System to populate delivery profiles. Rather, these are independent functions that happen to be based on the same user source data.

D.1.1 About Group and Application Role Resolution

In this release of Oracle Business Intelligence, application roles are used to define security policies rather than groups. When you create an agent, you can choose whether it should be delivered to a user, an application role, or a Catalog group.

However, to maintain backward compatibility with previous releases, SA System still uses the Group Name column in the SA System source table to determine the e-mail addresses for the application roles and catalog groups that are specified for agents. Because of this, the SA System subject area functions the same as it did in previous releases, even though the Oracle Business Intelligence security model has changed significantly in the current release.

Because the group membership in SA System is used to determine the list of recipients rather than the membership of either application roles or Catalog groups, users should not add members to Catalog groups. Alternatively, administrators can synchronize the application role and Catalog group memberships with SA System whenever the memberships are updated.

D.2 Setting Up the Data Source for the SA System Subject Area

In your external data source, create a table called User that contains columns that correspond to the various delivery options. In addition, you must ensure that every user and group is present in the data.

Table D–1 shows the columns that are required for the SA System subject area table. You must create the columns that are listed in the order shown. Any external schema that has the information in this table can be mapped to the SA System subject area.

Table D–1 Columns in the SA System User Table

Column	Data Type	Description
Logon	VARCHAR	The unique user ID of the user that logs on to the system. This cannot be null.
Display Name	VARCHAR	The full name of the user. This can be null.
Group Name	VARCHAR	The name of the group to which this user belongs. If a user belongs to multiple groups, then there should be one row for each group in the SA System table. This should not be null if any data access security is based on group membership.
Time Zone	VARCHAR	This column is currently not used and exists for future use. This should be null.
Language	VARCHAR	This column is currently not used and exists for future use. This should be null.
Locale	VARCHAR	This column is currently not used and exists for future use. This should be null.
Email	VARCHAR	The primary e-mail address for the user. This is a complete SMTP address such as joe.perez@example.com. This can be null.
Email Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, or L for low priority. For example, if high, normal, and low priority alerts are to be delivered to this device, then the field should be HNL. If only high and normal priority alerts are to be delivered, then the field should be HN. This field should not be null if the Email column is specified. This can be null if Email is null.
Email Type	VARCHAR	This field can be one of two text strings, HTML or text. Because most primary e-mail clients can read rich MIME content (HTML with embedded images), HTML is usually the best choice. Choose text to support legacy e-mail clients that can read only plain text e-mail. This field should not be null if the Email column is specified. This can be null if Email is null.
Cell Phone	VARCHAR	This field is the complete SMTP address for the cell phone device that receives text message alerts. For example, 1015551234@cellphoneprovider.com. Only text messages are sent to this device. This can be null.

Table D–1 (Cont.) Columns in the SA System User Table

Column	Data Type	Description
Cell Phone Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, and L for low priority. This field should not be null if the Cell Phone column is specified. This can be null if Cell Phone is null.
Pager	VARCHAR	This field is the complete SMTP address for the pager device that receives text message alerts. For example, 1015555678@pagerprovider.com. Only text messages are sent to this device. This can be null.
Pager Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, and L for low priority. This field should not be null if the Pager column is specified. This can be null if Pager is null.
Handheld	VARCHAR	This field is the complete SMTP address for the handheld device that receives text message alerts. For example, joe.perez@handheldprovider.com. Only text messages are sent to this device. This can be null.
Handheld Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, and L for low priority. This field should not be null if the Handheld column is specified. This can be null if Handheld is null.

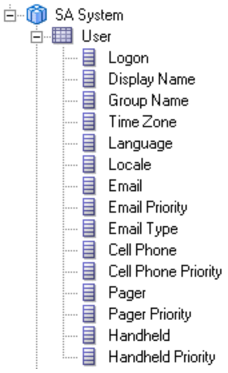
D.3 Importing SA System Data Into the Repository

After you configure the external data source, you must create and build the subject area in the Oracle BI repository. To do this, you first import the User table from the data source into the Physical layer. Then, map the User table and columns from the Physical layer to the Business Model and Mapping layer. Finally, map the User table and columns from the Business Model and Mapping layer to the Presentation layer. The name for the subject area must always be SA System.

See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about importing data into the repository and modeling information in the Business Model and Mapping layer and Presentation layer.

Figure D–1 shows the table and columns in the Presentation layer of the repository.

Figure D–1 SA System Subject Area in the Presentation Layer



- SA System
  - User
    - Logon
    - Display Name
    - Group Name
    - Time Zone
    - Language
    - Locale
    - Email
    - Email Priority
    - Email Type
    - Cell Phone
    - Cell Phone Priority
    - Pager
    - Pager Priority
    - Handheld
    - Handheld Priority

D.4 Setting Configuration Options for the SA System Subject Area

You can control the availability of the delivery options that are configured in the SA System subject area and the user-defined delivery options by including certain elements in the instanceconfig.xml file. These elements take effect only if the SA System subject area is being used.

Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To change configuration options for the SA System subject area:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"

2. Locate the sections in which you must add the following elements:

   - SystemSubjectArea: Specifies whether to recognize the delivery devices and deliver profiles that are configured in the SA System subject area:

     – **True**. Recognizes the delivery devices and delivery profiles that are configured in the SA System subject area and displays them on the My Account page. (Default)

     – **False**. Ignores the delivery devices and delivery profiles that are configured in the SA System subject area and does not display them on the My Account page.

     Include this element within the ServerInstance element.

   - IgnoreWebcatDeliveryProfiles: Specifies whether to ignore user-defined delivery devices and deliver profiles:

     – **True**. Ignores the user-defined delivery devices and delivery profiles and does not display them on the My Account page. (This means that users cannot create new delivery devices and delivery profiles.)

     – **False**. Recognizes the user-defined delivery devices and delivery profiles and displays them on the My Account page. (Default)

     Include this element within the Alerts element, which is itself included in the ServerInstance element.

   - UpperCaseRecipientNames: Specifies that only users whose user names are uppercase can have agents delivered to them.

For example, suppose that you have users with names of user_lowercase and USER_UPPERCASE. If you set the UpperCaseRecipientNames element to true, then agents are sent only to USER_UPPERCASE.

Include this element within the Alerts element, which is itself included in the ServerInstance element.

3. Include the elements and their ancestor elements as appropriate, as shown in the following example.

```
<ServerInstance>
  <SubjectAreaMetadata>
    <SystemSubjectArea>true</SystemSubjectArea>
  </SubjectAreaMetadata>
   <Alerts>
     <IgnoreWebcatDeliveryProfiles>false</IgnoreWebcatDeliveryProfiles>
     <UpperCaseRecipientNames>true</UpperCaseRecipientNames>
   </Alerts>
</ServerInstance>
```

4. Save your changes and close the file.

5. Restart Oracle Business Intelligence.

D.4.1 Managing the Case of Login Names for the SA System Subject Area

When the SA System subject area is used, login names are compared to the Logon column in the SA System subject area. By default, this comparison is case-sensitive. This means, for example, that a login of "Fred" does not match an SA System subject area entry of "fred." If the authentication method is case-sensitive, then this works fine because the login "fred," accepted at login, matches "fred" in the Logon column in the SA System subject area.

If the case of login names does not match, then invalid user name errors might result. To avoid this situation, ensure that the SA_SYSTEM Logon value and the LDAP user name value have the same case. For example, set the LOGON value in the SA_USER table to lowercase by including it within the Lower() function.

You can also use the UpperCaseRecipientNames element in the instanceconfig.xml file to ensure delivery only to users whose names are uppercase.

E

Advanced Configuration Reference

This appendix provides advanced postinstallation configuration and administration procedures that are *not* specific to analyses, agents, dashboards, or the Oracle BI Presentation Catalog. Directions for configuring these components of Oracle Business Intelligence are in earlier chapters. Most administrators need not change the configuration settings that are described in this appendix.

This appendix contains the following topics:

- Section E.1, "Making Advanced Configuration Changes for Oracle BI Presentation Services"
- Section E.2, "Using the JavaHost Service for Oracle BI Presentation Services"

E.1 Making Advanced Configuration Changes for Oracle BI Presentation Services

The Oracle BI Presentation Services process hosts most of the business logic of the Web server and provides the framework and interface for the presentation of business intelligence data to Web clients.

- Under Windows, the process is *sawserver.exe*
- Under UNIX, the process is *sawserver*

The instanceconfig.xml file stores the configuration settings that affect Oracle BI Presentation Services. Many configuration settings are available in Fusion Middleware Control and that is the preferred method for making configuration changes. If a particular settings is not available in Fusion Middleware Control, then you can change it using the instanceconfig.xml file. You should make changes directly in this file only to change default elements, such as the name of the Oracle BI Presentation Catalog, or override internal default settings, such as those related to caches.

Several entries are present in the instanceconfig.xml file by default, including the path to the Oracle BI Presentation Catalog, and the name of the Oracle Business Intelligence Server data source name used by Presentation Services to access Oracle BI Server.

> **Note:** If you have previously made configuration changes by modifying the Windows registry, then you should migrate those changes to the instanceconfig.xml. In the Windows registry, entries under the Common key remain valid.

You can use the instanceconfig.xml file to customize various aspects of your deployment. For additional customization options that you can make in XML files for

Presentation Services, see Section 22.5, "Customizing the User Interface Using XML Message Files."

The following procedure provides information on general configuration changes that you can make. Before you begin this procedure, ensure that you are familiar with the information in Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings".

To manually edit the settings for general configuration changes:

1. Open the instanceconfig.xml file for editing, as described in Section 3.6, "Where are Configuration Files Located?"
2. Locate the section in which you must add the elements that are described in Table E–1.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <CaseInsensitiveMode>False</CaseInsensitiveMode>
  <UnaccessedRunningTimeoutMinutes>5</UnaccessedRunningTimeoutMinutes>
  <Security>
    <AllowRememberPassword>False</AllowRememberPassword>
    <CookieDomain>value</CookieDomain>
    <CookiePath>/analytics</CookiePath>
    <HardenXSS>true</HardenXSS>
  </Security>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

Table E–1 Elements for General Configuration

Element	Description	Default Value
AllowRememberPassword	Specifies whether to allow the browser to save the password, using browser-specific password management software. If set to true, prompts the user to specify whether to save the password for future sign-ins.	False

Table E–1 (Cont.) Elements for General Configuration

Element	Description	Default Value
CaseInsensitiveMode	Specifies whether to use case sensitivity when performing tasks in Presentation Services that relate to case, such as sorting. The BI Server has the CASE_SENSITIVE_CHARACTER_COMPARISON parameter that controls whether the BI Server uses case-sensitivity. You should ensure that Presentation Services has the same case-sensitivity setting as the BI Server. To do so, set the CaseInsensitiveMode element to the corresponding value. If the values do not correspond, then you might see inconsistent results in operations such as sorting. For example, if CASE_SENSITIVE_CHARACTER_COMPARISON is set to On, then set CaseInsensitiveMode to its default value of False so that case-sensitivity is enabled in both places. For more information, see Section A.4.4, "CASE_SENSITIVE_CHARACTER_COMPARISON."	False
CookieDomain	Specifies domain information for a cookie that is sent to the browser.	NA
CookiePath	Specifies the domain path to which cookies apply.	/analytics
HardenXSS	For a description, see Section E.1.1, "HardenXSS Element."	True
UnaccessedRunningTimeoutMinutes	Specifies the time to elapse, in minutes, before an unattended analysis is canceled. An unattended analysis is one that has not been accessed in the number of minutes specified by this setting. The minimum value is 2. This element addresses the case where a user is editing an analysis and browses elsewhere, abandoning the analysis, at least temporarily. Do not set the value too small, however, as the user might return to the analysis.	5

E.1.1 HardenXSS Element

You use the HardenXSS element to secure Presentation Services against cross-site scripting (XSS). Securing against XSS escapes input in fields in Presentation Services and renders it as plain text. For example, an unscrupulous user can use an HTML field to enter a script that steals data from a page.

By default, Presentation Services is secured against XSS, which means that users cannot save content that is flagged as HTML; instead only users who have the appropriate privileges can save content that contains HTML. In a default installation, only administrative users have the privileges to save content that is later rendered as HTML. Note that in a secure environment, only resources (images) that are located in Presentation Services can be used. These images are referenced using a relative path prefixed with "fmap:" (for example, fmap:images/someimage.gif) and cannot be retrieved using a full URL (for example, http://www.example.com/images/someimage.gif).

For information on privileges, see Section C.2.3, "Managing Presentation Services Privileges."

E.2 Using the JavaHost Service for Oracle BI Presentation Services

The JavaHost service gives Presentation Services the ability to use functionality that is provided in Java libraries to support the following components:

- Graph generation
- SVG renderer (Apache Batik)
- Oracle BI Scheduler (Java tasks support)
- Oracle BI Publisher
- Advanced reporting
- URL Connect (Issues an HTTP request to another component)
- Integration Service Call (Used by the Oracle BI Server to execute Java code)

To configure the JavaHost service, you can manually edit the configuration elements for the service in its configuration file (config.xml), located in the *ORACLE_INSTANCE*\config\OracleBIJavaHostComponent\coreapplication_obijh*n* directory. See Table E–2 for a description of the elements. The elements are identified by their relative path starting from the JavaHost element.

The common sub-elements, such as InputStreamLimitInKB, do not apply to the MessageProcessor, Listener, or SSL loaders.

For information on manual editing, see Section 3.4, "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings"

Table E–2 JavaHost Service Configuration Elements in config.xml

Element	Description
Loaders	Contains the ListOfEnabledLoaders and Loader elements. These elements specify the components for the JavaHost service. Avoid editing the elements in the Loaders section.
Loaders/ListOfEnabledLoaders	Specifies the list of components (such as Oracle BI Scheduler and BI Publisher) to be enabled. If this element is missing from the file, then all Loaders are enabled. If the element has an empty value, then all loaders are disabled. Each component has a corresponding Loader element. The name of the component listed here must match the name that is specified in the corresponding Loader/Name element.
Loaders/Loader	Contains the following elements, which specify configuration information for a specific component: ■ Name ■ Class ■ ConfigNodePath ■ ClassPath
Loaders/Loader/Name	Specifies the unique name of the component. Use this name in the ListOfEnabledLoaders element.
Loaders/Loader/Class	Specifies the main class for the component.

Table E–2 (Cont.) JavaHost Service Configuration Elements in config.xml

Element	Description
Loaders/Loader/ConfigNodePath	Specifies the XPath (starting from the JavaHost element) to the configuration information for the Loader.
Loaders/Loader/ClassPath	Specifies the paths for the JAR files of libraries that the JavaHost service can use.
InputStreamLimitInKB	A sub-element common to each loader that specifies, in kilobytes, the maximum input size for requests that are sent to JavaHost. A value of zero deactivates this limit. If the maximum size is exceeded, then an error message is displayed. Default: 8192
RequestResponseLogDirectory	A sub-element common to each loader that specifies the name of the directory for the response files of requests. Default: A default temp directory
LogLargeRequests	A sub-element common to each loader that specifies whether to create a response file when processing large requests. Default: True
ReadRequestBeforeProcessing	A sub-element common to each loader that specifies whether to wait to process the request until a file is completely read. Default: True
LargeRequestThresholdInKB	A sub-element common to each loader that specifies, in kilobytes, the maximum size before using disk space for requests. For requests larger than this size, use disk space instead of memory to cache the requested data. The larger this value is the more memory that the JavaHost service might potentially use and the faster the request processing can occur. This setting also establishes the threshold for the LogLargeRequests element. Default: 200
MessageProcessor	Contains the SocketTimeout element.
MessageProcessor/SocketTimeout	Specifies the idle timeout (in milliseconds) for the socket, after which the socket is returned to the idle sockets pool. JavaHost uses a socket polling mechanism to wait for new data on the whole set of idle sockets in a single thread. Initial messages in the idle pool are handled through Java NIO channels. Default: 5000 (5 seconds)
Listener	Contains the following elements: ▪ PermittedClientList ▪ Port ▪ Address ▪ Secure
Listener/PermittedClientList	Specifies a list of IP addresses and host names from which JavaHost accepts incoming connections. Separate each client's IP address or host name by a comma. To accept all client connections, set this element to an asterisk (*). Default: *

Table E–2 (Cont.) JavaHost Service Configuration Elements in config.xml

Element	Description
Listener/Port	Identifies the JavaHost TCP/IP listening port. Default: 9810
Listener/Address	Specifies the network interface that JavaHost is to bind to. If this element has no value, then JavaHost binds to all available network interfaces.
Listener/Secure	Specifies whether to enable SSL encryption for the JavaHost service: ■ Yes: Enables SSL encryption ■ No: Disables SSL encryption Default: No For information on SSL, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
Batik	Contains only the common sub-elements such as InputStreamLimitInKB, as they relate to converting SVG graphics to rasterized image formats.
Scheduler	Contains the following elements: ■ Enabled ■ DefaultUserJarFilePath ■ DefaultTempFilePath ■ DefaultPurgingPeriod
Scheduler/Enabled	Specifies whether to enable the interaction of the JavaHost service with Oracle BI Scheduler: ■ True: Enables interaction with Oracle BI Scheduler ■ False: Disables interaction with Oracle BI Scheduler Default: False
Scheduler/DefaultUserJarFilePath	Specifies the default directory for storing JAR files for the Java extension utility. When Oracle BI Scheduler is enabled, this element is required and accepts a single path.
Scheduler/DefaultTempFilePath	Specifies the default directory for storing temporary files for Oracle BI Scheduler requests. Default: the system temp directory
Scheduler/DefaultPurgingPeriod	Specifies the default period (in seconds) for Oracle BI Scheduler requests to clean up failed jobs. Default: 300
XMLP	Contains only the common sub-elements such as InputStreamLimitInKB and ReadRequestBeforeProcessing, as they relate to Oracle BI Publisher.
URLConnect	Contains elements that relate to SSL. Avoid modifying these elements.
DVT	Contains only the common InputStreamLimitInKB sub-element by default, as they relate to graph generation. You can add other common sub-elements as necessary.

F

Propagating Labels and Tooltips from Oracle ADF Data Sources

You can propagate user interface (UI) hints (that is, labels and tooltips), from an Oracle Application Development Framework data source to display when users work with analyses. When translated labels and tooltips (based on user locale) are maintained within an Oracle ADF data source, you can query the data source to access this translated data. You use the Administration Tool to configure presentation columns to use when creating analyses.

This section contains the following topics:

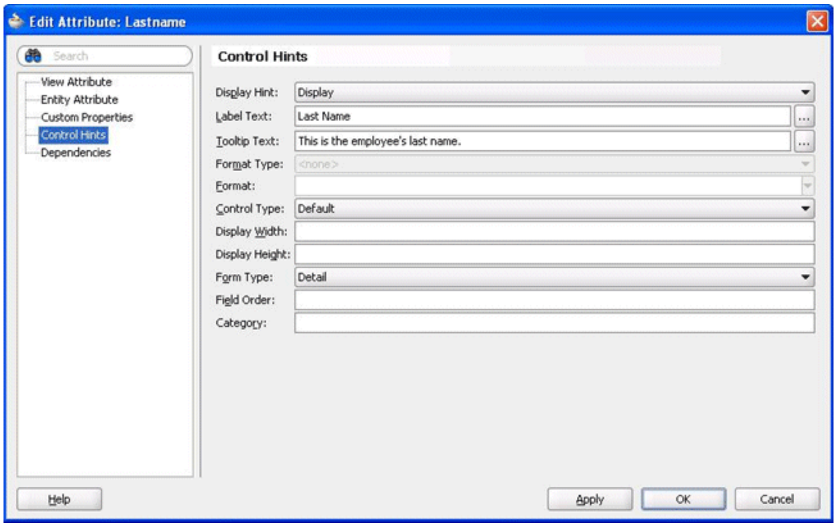
- Section F.1, "What are Labels and Tooltips?"
- Section F.2, "Session Variable Naming Scheme for UI Hints from an Oracle ADF Data Source"
- Section F.3, "Mapping of Presentation Columns to Physical Columns"
- Section F.4, "Initializing Session Variables Automatically for Propagating UI Hints"
- Section F.5, "Example: Using UI Hints From an Oracle ADF Data Source When Creating Analyses"
- Section F.6, "Using XML Code in Initialization Blocks to Query UI Hints from an Oracle ADF Data Source"

F.1 What are Labels and Tooltips?

A label is the text that is used in prompts or table headers that precedes the value of a data item. A tooltip is the text that is displayed when a user hovers the mouse pointer over the item. Each attribute of a view object (VO) has an associated label and tooltip. A view object is the Oracle Application Development Framework component that enables a developer to work easily with SQL query results. The propagation of UI hints enables a presentation column in the Administration Tool to use a label and tooltip as its Custom display name and Description respectively.

Figure F–1 shows the Label Text and Tooltip Text options in the Edit Attribute dialog in Oracle JDeveloper.

Figure F–1 Edit Attribute Dialog Box in JDeveloper for Label and Tooltip Options



Control Hints

Display Hint: Display

Label Text: Last Name

Tooltip Text: This is the employee's last name.

Format Type: <none>

Format:

Control Type: Default

Display Width:

Display Height:

Form Type: Detail

Field Order:

Category:

F.2 Session Variable Naming Scheme for UI Hints from an Oracle ADF Data Source

Session variable names are generated by the Oracle BI Enterprise Edition broker servlet in Oracle WebLogic Server in the following format:

ADF_*UI Hint Type*_*Database Name*_*VO's Name*_*Attribute's Name*

Where:

UI Hint Type is either LABEL or TOOLTIP, depending on the UI hint type that the session variable represents.

Database Name is the value of the "database" attribute of the ADFQuery element in the XML query. Special characters such single quotes ('), double quotes ("), and spaces are replaced by the underscore character.

VO's Name is the name of the View Object to which the attribute belongs. Oracle ADF prohibits special characters and spaces in the name.

Attribute's Name is the name of the attribute that the session variable represents. Oracle ADF prohibits special characters and spaces in the name.

Every character in the session variable name is uppercase. For example, the XML query in Example F–3 generates four session variables with the following names:

ADF_LABEL_MY_ORCLADF_EMPLOYEESVIEW_FIRSTNAME

ADF_TOOLTIP_MY_ORCLADF_EMPLOYEESVIEW_FIRSTNAME

ADF_LABEL_MY_ORCLADF_EMPLOYEESVIEW_LASTNAME

ADF_TOOLTIP_MY_ORCLADF_EMPLOYEESVIEW_LASTNAME

F.3 Mapping of Presentation Columns to Physical Columns

As required by the naming scheme for session variables, each presentation column must map to a physical column. When you select **Externalize Display Names > Generate ADF Label** or **Externalize Descriptions > Generate ADF Tooltip** for a presentation layer object, then the physical column is located using the following rules:

1. Examine the presentation column and determine its logical column. If the logical column is derived from an existing logical column, then the physical column cannot be found.
2. If the default aggregation rule for the logical column is not None or Sum, then the physical column cannot be found. It does not make sense semantically to use the ADF UI hints for aggregation rules other than Sum.
3. A logical column can be mapped to physical columns by multiple logical table sources. Only logical table sources that are not disabled are searched.
4. Do not search logical table sources that map the logical column using non-trivial expressions (that is, anything more than a physical column name). If no logical table sources are searched, then the physical column cannot be found.
5. From the remaining ordered list of logical table sources, examine the physical column that is mapped by the first logical table source. The physical column must be mapped to a VO attribute. In other words the physical column must be part of a physical database of type OracleADF11g.
   - If this condition is satisfied, then the physical column for obtaining UI hints is found.
   - If this condition is not satisfied, then continue to examine the physical column that is mapped by the next logical table source until the physical column that is mapped to a VO attribute is found.

     If all logical table source are searched without satisfying the condition, then the physical column cannot be found.

If the physical column for obtaining UI hints is found using these rules, then the custom display name or description is populated with the session variable whose name is based on the scheme that is described in Section F.2, "Session Variable Naming Scheme for UI Hints from an Oracle ADF Data Source."

If the physical column for obtaining UI hints is not found using these rules, then the **Generate ADF Label** and **Generate ADF Tooltip** options are shown as disabled in the right-click menu.

As an alternative to using the physical column found using these rules, you can use XML code in an initialization block to initialize your own session variables with ADF UI hints. You must then enter these session variable names in the **Custom display name** and **Custom description** fields manually. See Section F.6, "Using XML Code in Initialization Blocks to Query UI Hints from an Oracle ADF Data Source" for more information.

F.4 Initializing Session Variables Automatically for Propagating UI Hints

If the **Externalize Display Names > Generate ADF Label** and **Externalize Descriptions > Generate ADF Tooltip** options were used to successfully generate the session variable names for UI hints from Oracle ADF, then the session variables are created and initialized when Oracle BI Presentation Services queries them during the session. The variables are not created and initialized during the session logon stage for performance reasons. Instead, the variables are created and initialized when they are needed by a specific query within a session, using the **Allow deferred execution** feature.

When Presentation Services queries the custom display names and custom descriptions through ODBC, the Oracle BI Server checks if the associated session variables have been created. If they have not been created, then the BI Server dynamically generates the appropriate XML query (as described in Section F.6, "Using XML Code in Initialization Blocks to Query UI Hints from an Oracle ADF Data Source") to query the UI hints from the Oracle ADF data source. The BI Server uses the UI hints to create and initialize the session variables. As an optimization, the BI Server queries UI hints per VO; that is, if the BI Server needs the UI hints of a VO's attributes, then the UI hints for all the attributes under the VO are queried and propagated through session variables.

F.5 Example: Using UI Hints From an Oracle ADF Data Source When Creating Analyses

The following example shows how you can use UI hints from an Oracle ADF data source when creating analyses.

The following prerequisites must be met:

- UI hints must have been configured in the Oracle ADF data source.
- A working repository must have been configured for the Oracle ADF data source in the Administration Tool.

To use UI hints from an Oracle ADF data source when creating analyses:

1. Suppose that the repository contains a presentation column named "LastName." On the General tab of the Presentation Column dialog, the **Custom display name** and **Custom description** fields are not selected.

   Right-click the column in the Presentation layer and select first **Externalize Display Names > Generate ADF Label**, then **Externalize Descriptions > Generate ADF Tooltip** to generate the strings that populate the **Custom display name** and **Custom description** fields.

   You can also use these options from the right-click menu of a presentation table to generate the strings for all the columns in that table.

2. View the UI hints:

   a. Sign in to Oracle Business Intelligence.

   b. Create a new analysis using the subject area for which you obtained UI hints.

   c. In the Subject Areas pane, expand the Employee folder to see the UI hints that have been propagated from the Oracle ADF data source.

      The LastName column displays as "Last Name" (the label value from the Oracle ADF data source). When you hover the mouse pointer over the column, the tip displays as "This is the employee's last name" (the tooltip value from the Oracle ADF data source).

   For information about creating analyses, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

F.6 Using XML Code in Initialization Blocks to Query UI Hints from an Oracle ADF Data Source

As an alternative to using the automated system described in the previous section, you can use specialized XML code in place of SQL statements in initialization blocks to query the data source for UI hints, within a single repository and subject area. You use the ADFQuery element, which has three attributes that are named mode, database, and locale. The element requires zero or more child elements. The syntax of the element is as follows:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<ADFQuery mode="{Mode}" database="{Database Name}"
locale="VALUEOF(NQ_SESSION.WEBLANGUAGE)">
     <ViewObject><![CDATA[{VO Name}]]></ViewObject>
     <Attribute>
     <ViewObject><![CDATA[{Attribute VO Name}]]></ViewObject>
     <Name><![CDATA[{Attribute Name}]]></Name>
     </Attribute>
</ADFQuery>
```

where

{Mode} specifies what you want to query:

- `label` for querying attributes' label
- `tooltip` for querying attributes' tooltip
- `ui_hints` for querying attributes' label and tooltip

{Database Name}

Use the name of the physical database object in the Administration Tool, which contains the physical columns that correspond to the attributes in the Oracle ADF data source.

{VO Name}

Use the name of the View Object to obtain the UI hints of all attributes in it.

{Attribute VO Name}

Use the name of the View Object that contains the attribute.

{Attribute Name}

Use the name of the attribute that belongs to the associated View Object to obtain the UI hints of this attribute.

Example F–1 Querying Labels for All View Objects

No child elements must be included in the ADFQuery element, if the UI hints of all attributes in all View Objects are queried. For example, to query the labels of all attributes in all View Objects under the My_orclADF physical database object, use the following XML code:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<ADFQuery mode="label" database="My_orclADF"
locale="VALUEOF(NQ_SESSION.WEBLANGUAGE)">
</ADFQuery>
```

Example F–2 Querying Tooltips for Specific View Objects

The ADFQuery element can contain zero or more child elements named ViewObject if UI hints of all attributes in specific View Objects are queried. Each ViewObject element has a text content that contains the View Object's name. The ViewObject element is used to specify the View Objects from which the UI hints of all attributes are queried. For example, to query the tooltips of all attributes in the View Object that is named EmployeesView and CustomersView under the My_orclADF physical database object, use the following XML code:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
```

```
<ADFQuery mode="tooltip" database="My_orclADF"
locale="VALUEOF(NQ_SESSION.WEBLANGUAGE)">
   <ViewObject><![CDATA[EmployeesView]]></ViewObject>
   <ViewObject><![CDATA[CustomersView]]></ViewObject>
</ADFQuery>
```

Example F–3 Querying UI Hints for Specific Attributes

The ADFQuery element can contain zero or more child elements named Attribute. Each Attribute element has two required child elements named ViewObject and Name. The Attribute element is used to specify the attributes from which the UI hints are queried. The ViewObject child element has a text content that contains the View Object's name. This element specifies the View Object that the attribute belongs to. The Name child element has a text content which contains the attribute's name. For example, to query the labels and tooltips of the attributes named Firstname and Lastname in the EmployeesView View Object under the My_orclADF physical database object, use the following XML code:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<ADFQuery mode="ui_hints" database="My_orclADF"
locale="VALUEOF(NQ_SESSION.WEBLANGUAGE)">
   <Attribute>
      <ViewObject><![CDATA[EmployeesView]]></ViewObject>
      <Name><![CDATA[Firstname]]></Name>
   </Attribute>
   <Attribute>
      <ViewObject><![CDATA[EmployeesView]]></ViewObject>
      <Name><![CDATA[Lastname]]></Name>
   </Attribute>
</ADFQuery>
```

After configuring the initialization blocks, you must manually enter the session variable names in the **Custom display name** and **Custom description** text fields for the appropriate presentation column.

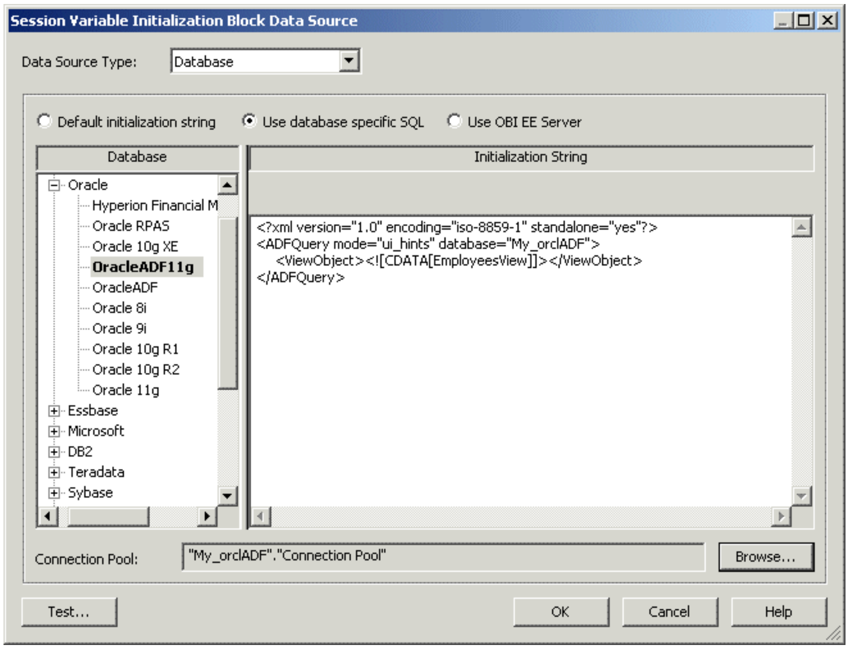
Follow the procedure in the example in Section F.5, "Example: Using UI Hints From an Oracle ADF Data Source When Creating Analyses", but replace the first step with the following ones:

1. Create session initialization blocks in the Administration Tool.

   a. In the Session Variable Initialization Block Data Source dialog, enter the Initialization string.

      In this example, the initialization block queries both the label and tooltip of all attributes in a View Object named EmployeesView. Figure F–2 shows the setup of a session variable initialization block with an appropriate Oracle ADF UI hint query. "My_orclADF"."Connection Pool" is a connection pool for an Oracle ADF data source.

Figure F–2 Setup of a Session Variable Initialization Block Data Source with an Oracle ADF UI Hints Query



Data Source Type: Database

○ Default initialization string ● Use database specific SQL ○ Use OBI EE Server

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<ADFQuery mode="ui_hints" database="My_orclADF">
   <ViewObject><![CDATA[EmployeesView]]></ViewObject>
</ADFQuery>
```

Connection Pool: "My_orclADF"."Connection Pool"

b. In the Session Variable Initialization Block dialog, select **Row-wise initialization** as the Variable Target.

c. Click **Test** to test the query against the Oracle ADF data source.

   In the results window, the first column contains the session variable names that are generated using the naming scheme that is described in Section F.2, "Session Variable Naming Scheme for UI Hints from an Oracle ADF Data Source." The second column contains the label and tooltip values from the Oracle ADF data source.

2. Configure a custom display name and a description in presentation columns.

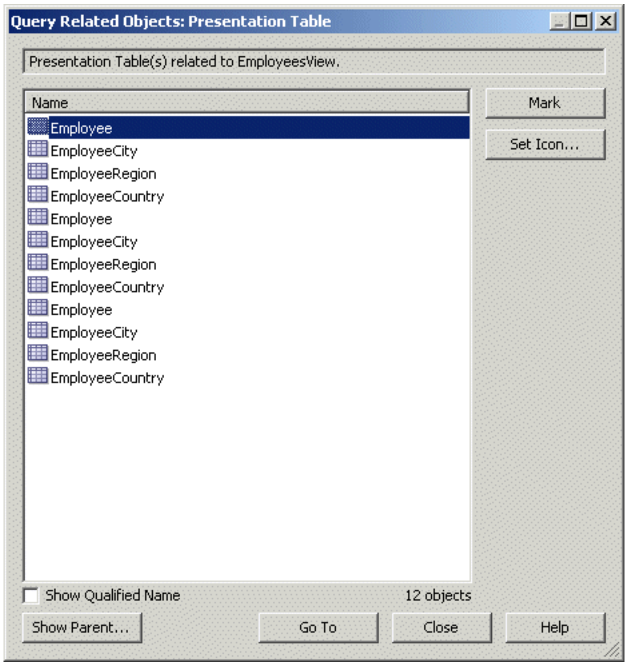
   To find the presentation tables that can use the UI hints from the EmployeesView View Object, this example uses the Query Repository feature in the Administration Tool.

   a. Right-click a physical table (for example, EmployeesView), then select **Query Related Objects** > **Presentation** > **Presentation Table** from the menu.

      The Query Related Objects dialog displays all the related presentation tables.

      This example sets up a custom display name and custom description for columns in the Employee presentation table.

Figure F–3 Using the Query Related Objects Feature to Find the Related Presentation Tables



Query Related Objects: Presentation Table

Presentation Table(s) related to EmployeesView.

Name

- Employee
- EmployeeCity
- EmployeeRegion
- EmployeeCountry
- Employee
- EmployeeCity
- EmployeeRegion
- EmployeeCountry
- Employee
- EmployeeCity
- EmployeeRegion
- EmployeeCountry

□ Show Qualified Name

12 objects

b. Select the required presentation table and click **Go To**.

This displays the selected presentation table.

c. Expand the presentation table to view the presentation columns.

d. Double-click the LastName presentation column to display the Presentation Column dialog.

e. Select **Custom display name** and enter a value such as the following one:

```
VALUEOF(NQ_SESSION.ADF_LABEL_MY_ORCLADF_EMPLOYEESVIEW_LASTNAME)
```

f. Select **Custom description** and enter a value such as the following one:

```
VALUEOF(NQ_SESSION.ADF_TOOLTIP_MY_ORCLADF_EMPLOYEESVIEW_LASTNAME)
```

g. Click **OK**.

h. Save the changes in the repository and restart the Oracle BI Server.

G

Mapping User Interface Labels with Configuration File Elements

This appendix maps Fusion Middleware Control User Interface (UI) labels for Oracle Business Intelligence with the corresponding element names used in configuration files. The information in the following tables is included here for completeness. You do not need this information for most operations.

Notes

- For information about elements for the Oracle BI Server that are not included in the following tables, see Appendix A, "NQSConfig.INI File Configuration Settings".
- For information about the location of configuration files, see Section 3.6, "Where are Configuration Files Located?"

Table G–1 Scalability Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File	Related Information
BI Servers	■ CLUSTER_PARTICIPANT ■ SERVERS	■ NQSConfig.INI ■ NQClusterConfig.INI	Section 5.5, "Using Fusion Middleware Control to Scale System Components"
Presentation Servers	Web_Server	instanceconfig.xml (for Oracle BI Scheduler)	Section 5.5
JavaHosts	■ JavaHostProxy ■ Java_Host_Server ■ JAVAHOST_HOSTNAME_OR_IP_ADDRESSES	■ instanceconfig.xml (for Presentation Services) ■ instanceconfig.xml (for Oracle BI Scheduler) ■ NQSConfig.INI	Section 5.5
Port Range From / Port Range To	■ RPC_SERVICE_OR_PORT MONITOR_PORT JAVAHOST_HOSTNAME_OR_IP_ADDRESSES ■ Listener ■ Web_Server PortString ClusterPort ■ Port	■ NQSConfig.INI ■ instanceconfig.xml (for Presentation Services) ■ instanceconfig.xml (for Oracle BI Scheduler) ■ config.xml	Section 5.5
Listen Address	Not directly mapped to a configuration element.	N/A	Chapter 5

Table G–2 Availability Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File	Related Information
Primary Host / Instance	■ PRIMARY_CONTROLLER ENABLE_CONTROLLER SCHEDULERS ■ Alerts	■ NQClusterConfig.INI ■ instanceconfig.xml (for Presentation Services)	Section 6.2.1, "Using Fusion Middleware Control to Configure Primary and Secondary Instances"
Secondary Host / Instance	■ SECONDARY_CONTROLLER SCHEDULERS ■ Alerts	■ NQClusterConfig.INI ■ instanceconfig.xml (for Presentation Services)	Section 6.2.1

Table G–3 Performance Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File (NQSConfig.INI for the BI Server and instanceconfig.xml for Presentation Services)	Related Information
Cache enabled	ENABLE	NQSConfig.INI	Section 7.5.1, "Using Fusion Middleware Control to Enable and Disable Query Caching"
Maximum cache entry size	MAX_CACHE_ENTRY_SIZE	NQSConfig.INI	Section 7.5.2, "Using Fusion Middleware Control to Set Query Cache Parameters"
Maximum cache entries	MAX_CACHE_ENTRIES	NQSConfig.INI	Section 7.5.2
Global cache path	GLOBAL_CACHE_STORAGE_PATH	NQSConfig.INI	Section 7.5.4, "Using Fusion Middleware Control to Set Global Cache Parameters"
Global cache size	GLOBAL_CACHE_STORAGE_PATH	NQSConfig.INI	Section 7.5.4, "Using Fusion Middleware Control to Set Global Cache Parameters"
Disallow Online RPD Updates	READ_ONLY_MODE	NQSConfig.INI	Section 7.3.1, "Using Fusion Middleware Control to Disallow RPD Updates"
User Session Expiry	ClientSessionExpireMinutes	instanceconfig.xml	Section 7.3.2, "Using Fusion Middleware Control to Set the User Session Log-Off Period"
Maximum Number of Rows Processed to Render a Table View	ResultRowLimit	instanceconfig.xml	Section 7.3.4, "Using Fusion Middleware Control to Set the Maximum Number of Rows Processed to Render a Table"
Maximum Number of Rows to Download to Excel	DefaultRowsDisplayedInDownload	instanceconfig.xml	Section 7.3.3, "Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables"
Maximum Number of Rows Per Page to Include in Email	DefaultRowsDisplayedInDelivery	instanceconfig.xml	Section 7.3.3, "Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables"

Table G–4 Log Configuration Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File	Related Information
Maximum File Size	maxFileSizeKb	instanceconfig.xml (for Presentation Services and BI Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.1.2.1, "Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels"
Maximum Log Age	MaximumLogAgeDay	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.1.2.1
Maximum File Size **Note:** Field in Query Logs region.	MaximumFileSizeKb	logconfig.xml (for the BI Server)	Section 8.1.2.1
Maximum Log Age **Note:** Field in Query Logs region.	MaximumLogAgeDay	logconfig.xml (for the BI Server)	Section 8.1.2.1
Incident Error	IncidentError	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.1.2.1
Error	Error	instanceconfig.xml (for Presentation Servicesand Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.1.2.1

Table G–4 (Cont.) Log Configuration Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File	Related Information
Warning	Warning	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.1.2.1
Notification	Notification	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.1.2.1
Trace	Trace	instanceconfig.xml (for Presentation Servicesr and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.1.2.1

For information about diagnostic log configuration files (for example, logconfig.xml), see:

- Section 8.2.2, "What Are Diagnostic Log Configuration Files and Where are They Located?"
- Section 8.2.3, "What Are Log File Message Categories and Levels?"
- Section 8.2.4, "What is Log File Rotation?"

Table G–5 Security Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File	Related Information
Enable SSO	EnabledSchemas (indirectly associated)	instanceconfig.xml	*Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
SSO Provider	EnabledSchemas (indirectly associated)	instanceconfig.xml	*Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*

Table G–6 Presentation Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label Name	Configuration Element	Configuration File	Related Information
Show page tabs	ShowPageTabsAlways	instanceconfig.xml	Section 12.1, "Using Fusion Middleware Control to Change Presentation Setting Defaults"
Show section headings	ShowSectionHeadingsDefault	instanceconfig.xml	Section 12.1
Allow dashboard sections to be collapsible	CollapsibleSectionsDefault	instanceconfig.xml	Section 12.1
Pivot Tables show auto-preview	DisableAutoPreview	instanceconfig.xml	Section 12.1

Table G–7 Repository Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File (NQSConfig.INI is for BI Server and instanceconfig.xml is for Presentation Services)	Related Information
Default RPD	Maps to the versioned name of the current RPD file.	NQSConfig.INI	Appendix A.2, "Repository Section Parameters"
Password	Not mapped to a configuration file element, but accessible from the Credentials page in Fusion Middleware Control: 1. In Fusion Middleware Control, under the WebLogic Domain node, select **bifoundation_domain**. 2. From the WebLogic Domain menu, select **Security** and then select **Credentials**. 3. On the Credentials page, expand **oracle.bi.enterprise** and select an entry of the format repository.*rpd_name*.	N/A	Section 10.1, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location" *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
Repository File	Repository section, default repository entry Note that when you browse to upload an RPD file, the path name is not written to a file. Instead, the uploaded file creates the configuration file entry as follows: *logical_name* = *repository_name*.rpd, DEFAULT; For example, Star = SampleAppLite.rpd, DEFAULT; where Star is the DSN, and SampleAppLite.rpd is the name of the RPD file.	NQSConfig.INI	Section 10.1

Table G–7 (Cont.) Repository Tab - Mapping Between User Interface Labels and Configuration File

Fusion Middleware Control UI Label	Configuration Element	Configuration File (NQSConfig.INI is for BI Server and instanceconfig.xml is for Presentation Services)	Related Information
Share Repository	REQUIRE_PUBLISHING_DIRECTORY	NQSConfig.INI	Section 10.1
Shared Location	REPOSITORY_PUBLISHING_DIRECTORY	NQSConfig.INI	Section 10.1
Catalog Location	CatalogPath	instanceconfig.xml	Section 10.1

Table G–8 Scheduler Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File (for BI Scheduler)	Related Information
Database	TargetType	instanceconfig.xml	Section 25.2.1, "Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler"
Call Interface	CLIType	instanceconfig.xml	Section 25.2.1
Data Source	DSN	instanceconfig.xml	Section 25.2.1
Username	scheduler.schema	Credential found in oracle.bi.enterprise credential map.	Section 25.2.1 *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
Password	See Username.	N/A	Section 25.2.1
Confirm Password	See Username.	N/A	Section 25.2.1

Table G–9 Marketing Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File	Related Information
URL	Marketing, ContentServer, URL	instanceconfig.xml for Presentation Services	Section 11.3.1, "Using Fusion Middleware Control to Configure the Connection to the Marketing Content Server"
Username	marketing.content.server	Credential found in oracle.bi.enterprise credential map.	Section 11.3.1 *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
Password	See Username.	N/A	Section 11.3.1
Confirm Password	See Username.	N/A	Section 11.3.1

Table G–10 Mail Tab - Mapping Between User Interface Labels and Configuration File Elements

Fusion Middleware Control UI Label	Configuration Element	Configuration File (for BI Scheduler)	Related Information
SMTP Server	SMTP_Server	instanceconfig.xml	Section 11.1.1, "Using Fusion Middleware Control to Configure Oracle BI Scheduler E-mail Settings That Affect Agents"
Port	SMTP_Port	instanceconfig.xml	Section 11.1.1
Display name of sender	From	instanceconfig.xml	Section 11.1.1
Email address of sender	Sender	instanceconfig.xml	Section 11.1.1
Username	mail.server	Credential found in oracle.bi.enterprise credential map	Section 11.1.1 *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
Password	See Username, above.	N/A	Section 11.1.1
Confirm password	See Username, above.	N/A	Section 11.1.1
Number of retries upon failure	Try	instanceconfig.xml	Section 11.1.1
Maximum recipients	MaxRecipients	instanceconfig.xml	Section 11.1.1
Addressing method	UseBcc	instanceconfig.xml	Section 11.1.1
Use SSL to connect to mail server	UseSSL	instanceconfig.xml	Section 11.1.1
Specify CA certificate source	This controls whether to fill in either SmtpCACertificateDir or SmtpCACertificateFile	instanceconfig.xml	Section 11.1.1
CA certificate directory	SmtpCACertificateDir	instanceconfig.xml	Section 11.1.1
CA certificate file	SmtpCACertificateFile	instanceconfig.xml	Section 11.1.1
SSL certificate verification depth	SmtpCertificateVerificationDepth	instanceconfig.xml	Section 11.1.1
SSL cipher list	SmtpCipherList	instanceconfig.xml	Section 11.1.1

H

Manually Updating Fusion Middleware Control Settings

This appendix describes how to manually update configuration settings that are normally managed by Fusion Middleware Control. The appendix contains the following topics:

- Section H.1, "Disabling Fusion Middleware Control from Managing Oracle Business Intelligence Configuration Settings"
- Section H.2, "Manually Updating Oracle Business Intelligence Configuration Settings Normally Managed by Fusion Middleware Control"

> **Note:** Use Fusion Middleware Control to update Oracle Business Intelligence configuration settings rather than manually updating configuration files using a text editor. For information, see Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings".

H.1 Disabling Fusion Middleware Control from Managing Oracle Business Intelligence Configuration Settings

> **Caution:** Do not attempt to manually update configuration settings that are normally managed by Fusion Middleware Control unless you have been specifically instructed by Oracle Support Services.

Complete this task to prevent manual updates from being overwritten by Fusion Middleware Control for Oracle Business Intelligence configuration settings that are normally managed by Fusion Middleware Control.

You should not manually update configuration settings (that are normally managed by Fusion Middleware Control) without disabling Fusion Middleware Control. Otherwise, your manual changes are overwritten when the Administration Server is restarted, or if changes are activated.

When Fusion Middleware Control is disabled, the Oracle Business Intelligence configuration settings are disabled in the user interface, the **Lock Configuration** buttons are unavailable, and a message ("Central Configuration is disabled") is displayed.

If you disable Fusion Middleware Control from managing Oracle Business Intelligence configuration settings, then you can continue to use Fusion Middleware Control for other functions such as starting, stopping, restarting, monitoring performance, and viewing diagnostics.

You can disable Fusion Middleware Control from managing Oracle Business Intelligence configuration settings by manually updating a setting in the biee-domain.xml file, as described in the following procedure.

To disable Fusion Middleware Control from managing Oracle Business Intelligence configuration settings:

> **Caution:** Do not attempt to manually update a configuration setting that is normally managed by Fusion Middleware Control unless you have been specifically instructed by Oracle Support Services.

1. Stop Oracle Business Intelligence.

   For information, see Chapter 4, "Starting and Stopping Oracle Business Intelligence."

2. Complete the following steps to set the centralConfigurationEnabled setting in the biee-domain.xml file:

   a. Open the biee-domain.xml file in a text editor.

      The biee-domain.xml file is located in:

      user_projects\domains\bifoundation_domain\config\fmwconfig\biee-domain.xml

   b. Locate the centralConfigurationEnabled element and ensure that its value is set to false.

      To re-enable the element, set it to true.

   c. Save the biee-domain.xml file.

3. Restart Oracle Business Intelligence.

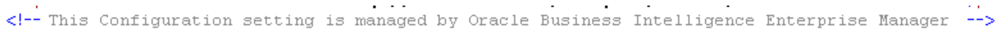
H.2 Manually Updating Oracle Business Intelligence Configuration Settings Normally Managed by Fusion Middleware Control

Do not disable Fusion Middleware Control from updating Oracle Business Intelligence configuration settings unless specifically instructed to by Oracle Support Services. If you must update configuration settings normally managed by Fusion Middleware Control, then you can use this procedure.

Configuration settings that are normally managed by Fusion Middleware Control, and which you manually update in a text editor, are overwritten by Fusion Middleware Control when you restart the Administration Server. Therefore you must disable Fusion Middleware Control from overwriting manual configuration changes for parts of the Oracle Business Intelligence system that are centrally managed.

> **Caution:** If you manually update configuration settings that are normally managed by Fusion Middleware Control (when Fusion Middleware Control is disabled), then you are likely to have to make identical changes in multiple copies of configuration files on multiple computers. If you fail to make appropriate changes in all of the necessary configuration files, then the system is incorrectly configured.

Configuration elements that are normally managed by Fusion Middleware Control are clearly marked in the configuration file as follows:



```
<!-- This Configuration setting is managed by Oracle Business Intelligence Enterprise Manager -->
```

To manually update the Oracle Business Intelligence configuration settings that are normally managed by Fusion Middleware Control:

1. If not yet done, disable Fusion Middleware Control from being able to update Oracle Business Intelligence configuration settings.

   For information, see Section H.1, "Disabling Fusion Middleware Control from Managing Oracle Business Intelligence Configuration Settings."

2. Stop Oracle Business Intelligence.

   For information, see Chapter 4, "Starting and Stopping Oracle Business Intelligence."

3. Make a backup copy of each configuration file that you plan to edit.

4. Open the configuration file in a text editor.

   > **Note:** If Oracle Business Intelligence is installed on multiple computers or in multiple locations, then a copy of the configuration file exists for each installation, and each file contains duplicate settings. Therefore, **you must locate and make the same changes in each copy of a configuration file**.

   For information, see Section 3.6, "Where are Configuration Files Located?"

5. In each configuration file to change, locate the element, or create the new element if a setting must be added to the file.

   For information, see Appendix G, "Mapping User Interface Labels with Configuration File Elements."

6. Enter the appropriate changes.

7. Save your changes and close the configuration file.

8. Restart Oracle Business Intelligence.

   For information, see Section 4.1, "About Starting and Stopping Oracle Business Intelligence."

Glossary

This glossary defines terms for Oracle Business Intelligence Enterprise Edition. See also the Oracle Fusion Middleware Master Glossary for additional terms and definitions.

action

Provides functionality to navigate to related content or to invoke operations, functions or processes in external systems. You can include actions in analyses, dashboard pages, agents, scorecard objectives, scorecard initiatives, and KPIs.

See also **action link**.

Action Framework

The Action Framework is a component of the Oracle BI EE architecture and includes a J2EE application called the Action Execution Service (AES) and actions-specific JavaScript functionality deployed as part of Oracle BI EE. The action framework also includes client-side functionality for creating actions and invoking certain action types directly from the browser.

action link

A link to an action that you have embedded in an analysis, dashboard page, scorecard objective, scorecard initiative, or KPI that, when clicked, runs an associated action.

See also **action**.

ADF Business Intelligence Component

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

Admin Server

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

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

Administration Tool

See **Oracle BI Administration Tool**.

agent

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

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

aggregate persistence

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

aggregate table

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

aggregation rule

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

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

alias table

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

analysis

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

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

analysis criteria

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

See also **analysis**.

analysis prompt

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

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

attribute

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

attribute column

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

See **hierarchical column**.

BI domain

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

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

See also **BI instance**.

BI instance

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

See also **BI domain**.

BI object

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

BI Search

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

bookmark link

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

See **prompted link**.

bridge table

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

briefing book

See **Oracle BI Briefing Books**.

business model

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

Business Model and Mapping layer

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

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

business owner

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

catalog

See **Oracle BI Presentation Catalog**.

cause & effect map

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

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

chronological key

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

Cluster Controller

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

column

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

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

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

column filter

See **filter**.

column prompt

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

See also **prompt**.

complex join

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

condition

Objects that return a single Boolean value based on the evaluation of an analysis or of a key performance indicator (KPI). You use conditions to determine whether agents deliver their content and execute their actions, whether actions links are displayed in dashboard pages, or whether sections and their content are displayed in dashboard pages.

See also **action**, **action link**, **agent** and **key performance indicator (KPI)**.

connection pool

An object in the Physical layer of an Oracle BI repository that contains the connection information for a data source.

See also **Physical layer**.

content designer

The user who creates business intelligence objects such as analyses, dashboards, and scorecards.

contextual event action

A pre-delivered action that utilizes the Action Framework to pass content from the business intelligence object to another region on an ADF page.

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

criteria

See **analysis criteria**.

cube

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

currency prompt

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

See also **prompt**.

custom view

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

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

dashboard

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

dashboard prompt

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

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

Dashboard URL

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

data source name (DSN)

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

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

database hint

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

dimension

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

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

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

See also **hierarchy**.

dimension table

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

See also **fact table**.

driving table

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

DSN

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

event polling table

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

Essbase

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

fact table

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

See also **dimension table**.

filter

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

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

foreign key

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

fragmentation content

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

Fusion Middleware Control

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

global header

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

Go URL

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

hierarchical column

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

See also **attribute column**.

hierarchy

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

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

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

hierarchy level

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

See also **presentation level**.

home page

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

image prompt

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

See also **prompt**.

initialization block

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

initiative

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

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

Java components

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

See also **Node Manager**.

key performance indicator (KPI)

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

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

KPI watchlist

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

See also **key performance indicator (KPI)**.

level

See **hierarchy level**.

logical display folder

Folders used to organize objects in the Business Model and Mapping layer. They have no metadata meaning.

logical join

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

logical layer

See **Business Model and Mapping layer**.

logical level

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

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

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

Logical SQL

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

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

logical table

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

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

logical table source

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

Managed Server

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

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

measure column

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

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

metadata

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

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

metadata dictionary

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

mission statement

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

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

multi-database join

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

Node Manager

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

See also **Java components**.

OCI

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

ODBC

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

object properties

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

See also **permissions**.

objective

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

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

offline mode

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

online mode

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

opaque view

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

Open Database Connectivity (ODBC)

A standard interface used to access data in both relational and non-relational databases. Database applications can use ODBC to access data stored in different types of database management systems, even if each database uses a different data storage format and programming interface.

OPMN

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

Oracle BI Administration Tool

A Windows application that is used to create and edit Oracle BI repositories. The Administration Tool provides a graphical representation of the three parts of a repository: the Physical layer, Business Model and Mapping layer, and the Presentation layer.

Oracle BI Briefing Books

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

Oracle BI JavaHost

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

Oracle BI Logical SQL View Object

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

Oracle BI Presentation Catalog

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

Oracle BI Presentation Services

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

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

Oracle BI Presentation Services server

The Oracle BI Web server that exchanges information and data with the Oracle BI Server.

Oracle BI Publisher

A J2EE application that provides enterprise-wide publishing services in Oracle Business Intelligence. It generates highly formatted, pixel-perfect reports.

See also **report**.

Oracle BI Publisher report

See **report**.

Oracle BI repository

A file that stores Oracle Business Intelligence metadata. The metadata defines logical schemas, physical schemas, physical-to-logical mappings, aggregate table navigation, and other constructs. The repository file has an extension of .rpd. Oracle BI repositories can be edited using the Oracle BI Administration Tool.

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

Oracle BI Scheduler

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

See also **results**.

Oracle BI Server

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

Oracle BI Server XML API

Provides utilities to create a generic, XML-based representation of the Oracle BI repository metadata. This XML file version of the repository can be used to programmatically modify the metadata. The Oracle BI Server XML API objects correspond to metadata repository objects in an RPD file. These objects are not the same as Oracle BI Presentation Catalog XML objects.

Oracle Business Intelligence Web Services

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

Oracle Business Intelligence Session-Based Web Services

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

See also Oracle Business Intelligence Web Services for SOA.

Oracle Business Intelligence Web Services for SOA

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

See also Oracle Business Intelligence Session-Based Web Services.

Oracle Call Interface (OCI)

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

Oracle Process Manager and Notification Server (OPMN)

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

See also **System components**.

Oracle Scorecard and Strategy Management

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

Oracle Technology Network (OTN)

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

parent-child hierarchy

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

See also **dimension**.

parent-child relationship table

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

pass-through calculation

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

permissions

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

See **object properties**.

perspective

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

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

physical catalog

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

physical display folder

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

physical join

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

Physical layer

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

physical schema

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

physical table

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

See also **Physical layer**.

presentation hierarchy

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

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

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

Presentation layer

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

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

presentation level

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

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

Presentation Services

See **Oracle BI Presentation Services**.

Presentation Services server

See **Oracle BI Presentation Services server**.

presentation table

An object in the Presentation layer of an Oracle BI repository that is used to organize columns into categories that make sense to the user community. A presentation table can contain columns from one or more logical tables. The names and object properties of the presentation tables are independent of the logical table properties.

primary key

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

process instance

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

See also **BI instance**.

prompt

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

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

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

prompted link

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

See **bookmark link**.

query

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

query cache

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

ragged hierarchy

See **unbalanced hierarchy**.

report

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

See also **analysis**.

repository

See **Oracle BI repository**.

repository variable

See **variable**.

results

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

See also **analysis**.

scorecard

See **Oracle Scorecard and Strategy Management**.

selection step

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

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

session variable

See **variable**.

skip-level hierarchy

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

See also **hierarchy**.

snowflake schema

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

SQL

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

star schema

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

strategy map

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

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

strategy tree

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

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

structured query language (SQL)

A standard programming language for querying and modifying data. Oracle Business Intelligence supports standard SQL-92 with several value-added proprietary extensions.

See also **Logical SQL**.

subject area

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

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

System components

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

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

transformation

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

unbalanced hierarchy

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

See also **hierarchy**.

value hierarchy

See **parent-child hierarchy**.

variable

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

Variables are of the following types:

- Repository variables have a single value at any point in time. There are two types of repository variables: static and dynamic.
- Session variables are created and assigned a value when each user logs on. There are two types of session variables: system and nonsystem.

variable prompt

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

See also **prompt**.

virtual physical table

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

vision statement

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

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

WebLogic domain

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

WebLogic Scripting Tool (WLST)

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

XML API

See **Oracle BI Server XML API**.

Index