

Oracle® Fusion Middleware

System Administrator's Guide for Oracle Business Intelligence
Enterprise Edition

11g Release 1 (11.1.1)

E10541-02

April 2011

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

E10541-02

Copyright © 2010, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: Christine Jacobs

Contributing Authors: Marla Azriel, Nick Fry, Dona Hobin

Contributors: Oracle Business Intelligence development, product management, and quality assurance teams

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xix
Audience	xix
Documentation Accessibility	xix
Related Documents	xx
Conventions	xx
New Features for Oracle Business Intelligence System Administrators	xxi
New Features for Oracle BI EE 11g Release 1 (11.1.1.5)	xxi
New Features for Oracle BI EE 11g Release 1 (11.1.1.3)	xxi
Part I Administering Oracle Business Intelligence	
1 Introduction to Oracle Business Intelligence System Administration	
1.1 What Are the Oracle Business Intelligence System Administration Tasks?	1-1
1.2 Getting Started with Managing Oracle Business Intelligence	1-2
1.3 What Is the Oracle Business Intelligence System Logical Architecture?	1-3
1.3.1 Oracle Business Intelligence on a Single Host	1-3
1.3.1.1 A Single Host and Enterprise Install	1-4
1.3.1.2 A Single Host and Simple Install	1-5
1.3.2 Oracle Business Intelligence on Multiple Hosts	1-6
1.3.3 About Java Components and System Components for Oracle Business Intelligence	1-6
1.3.3.1 About Java Components	1-6
1.3.3.2 About System Components	1-7
1.3.4 What Is the Administration Server?	1-7
1.4 What System Administration Tools Manage Oracle Business Intelligence?	1-8
1.4.1 Fusion Middleware Control	1-8
1.4.2 Oracle WebLogic Server Administration Console	1-9
1.4.3 Oracle Process Manager and Notification Server (OPMN) Tool	1-9
1.4.4 Oracle Weblogic Scripting Tool (WLST)	1-10
1.4.5 Oracle BI Administration Tool	1-10
1.4.6 Job Manager	1-10
1.5 Working with the Sample Application	1-10
1.6 Oracle BI Publisher Integration	1-10
1.7 Topics of Interest in Other Guides	1-10

1.8	System Requirements and Certification	1-11
-----	---	------

2 Managing Oracle Business Intelligence

2.1	Why Use Fusion Middleware Control and WebLogic Server Administration Console?	2-1
2.2	Centrally Managing Oracle Business Intelligence Components Using Fusion Middleware Control	2-2
2.2.1	Logging into Fusion Middleware Control to Manage Oracle Business Intelligence.	2-2
2.2.2	Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components	2-4
2.2.3	Using Fusion Middleware Control to Monitor Status and Start and Stop Oracle Business Intelligence Java Components	2-5
2.2.4	About the Navigation Tree in Fusion Middleware Control.....	2-7
2.3	Centrally Managing Oracle Business Intelligence Java Components Using the Oracle WebLogic Server Administration Console.....	2-7
2.4	Managing Oracle Business Intelligence JMX MBeans Using the Fusion Middleware Control MBean Browser	2-9
2.4.1	What Is the Fusion Middleware Control MBean Browser?	2-10
2.4.2	Displaying the Fusion Middleware Control MBean Browser.....	2-10

3 Configuring the Oracle Business Intelligence System

3.1	Configuring Oracle Business Intelligence	3-1
3.2	Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings	3-2
3.3	Using the Oracle BI Systems Management API to Update Configuration Settings.....	3-3
3.4	Using a Text Editor to Update Configuration Settings	3-3
3.5	Using the Fusion Middleware Control MBean Browser to Update Configuration Settings	3-4
3.6	Where Are Configuration Files Located?	3-5

Part II Managing Processes and Components

4 Starting and Stopping Oracle Business Intelligence

4.1	About Starting and Stopping Oracle Business Intelligence.....	4-1
4.2	Starting the Oracle Business Intelligence System.....	4-2
4.2.1	Starting Oracle Business Intelligence on Windows Platforms.....	4-2
4.2.2	Starting Oracle Business Intelligence on UNIX Platforms.....	4-3
4.3	Using Fusion Middleware Control to Start and Stop Oracle Business Intelligence System Components and Java Components.....	4-4
4.4	Using Oracle WebLogic Server Administration Console to Start and Stop Java Components	4-5
4.5	Alternative Methods for Starting and Stopping System Components	4-6
4.5.1	Using the OPMN Command Line to Start, Stop, Restart, and View the Status of System Components.....	4-6
4.5.2	Using a Windows Service to Start and Stop System Components	4-8
4.5.3	Using the Oracle BI Systems Management API to Programmatically Start and Stop Oracle Business Intelligence.....	4-9
4.6	Confirming Whether the Administration Server Is Running.....	4-9

4.7	Starting and Stopping the Administration Server and All Oracle Business Intelligence Java Components on Windows and UNIX	4-9
4.8	Starting and Stopping Managed Servers on Windows and UNIX	4-11
4.8.1	Confirming If the Managed Server Is Running and Starting It	4-12
4.9	Troubleshooting System Startup	4-13
4.9.1	Administration Server Fails to Start When the Database Is Not Running	4-13
4.9.2	Managed Server Is Down	4-14
4.9.3	Oracle BI Server Fails to Start.....	4-14
4.9.4	Oracle BI Presentation Services Fails to Start	4-14
4.9.5	Cannot Log In.....	4-14

Part III Scaling and Deploying for High Availability and Performance

5 Scaling Your Deployment

5.1	About Scaling Oracle Business Intelligence	5-1
5.1.1	How Do I Know When to Scale Out Processes?.....	5-2
5.1.2	What Processes Should I Scale?	5-2
5.2	Vertically Scaling Oracle Business Intelligence	5-3
5.3	Horizontally Scaling Oracle Business Intelligence.....	5-3
5.4	Setting Up Shared Files and Directories	5-4
5.4.1	Uploading and Sharing the Oracle BI Repository	5-4
5.4.2	Sharing the Oracle BI Presentation Catalog.....	5-5
5.4.3	Setting Up the Global Cache	5-5
5.4.4	Setting the Scheduler Script Path and Default Script Path	5-5
5.5	Using Fusion Middleware Control to Scale System Components.....	5-6
5.6	Validating That Your System Has Been Scaled Correctly	5-7
5.6.1	Using Fusion Middleware Control to View System Component Availability.....	5-8
5.6.2	Using the Administration Console to View Managed Server Availability.....	5-8

6 Deploying Oracle Business Intelligence for High Availability

6.1	About Oracle Business Intelligence Components in a Clustered Environment	6-2
6.1.1	Recommendations for Availability	6-3
6.1.2	Using Fusion Middleware Control to Identify Single Points of Failure.....	6-3
6.1.3	Achieving High Availability Using an Active-Passive Model.....	6-4
6.2	Configuring Oracle Business Intelligence Components for High Availability	6-4
6.2.1	Using Fusion Middleware Control to Configure Primary and Secondary Instances	6-5
6.3	Optional Configuration for Oracle Business Intelligence High Availability	6-6
6.3.1	Setting Optional Cluster Controller Parameters	6-6
6.3.2	Setting Optional Presentation Services Parameters.....	6-7
6.3.3	Setting Optional Oracle BI Presentation Services Plug-in Parameters	6-7
6.4	Using the Cluster Manager.....	6-8
6.4.1	Viewing and Managing Cluster Information	6-9
6.4.1.1	Status Information	6-9
6.4.1.2	Cache Information	6-10
6.4.1.3	Session Information.....	6-11
6.4.1.4	Server Information	6-13

6.5	Troubleshooting an Oracle Business Intelligence Clustered Environment.....	6-13
6.5.1	Avoiding Errors with Network Appliance Devices When the Oracle BI Server Is Running on Linux or UNIX.....	6-13

7 Managing Performance Tuning and Query Caching

7.1	Monitoring Service Levels	7-1
7.1.1	Using Fusion Middleware Control to View Common Performance Metrics	7-2
7.1.2	Using Fusion Middleware Control to View All Oracle Business Intelligence Metrics	7-3
7.1.3	Using the Administration Console to View Metrics for Java Components	7-3
7.2	About Query Performance Tuning.....	7-4
7.3	Setting Performance Parameters in Fusion Middleware Control.....	7-4
7.3.1	Using Fusion Middleware Control to Disallow RPD Updates	7-5
7.3.2	Using Fusion Middleware Control to Set the User Session Log-Off Period	7-5
7.3.3	Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables	7-6
7.3.4	Using Fusion Middleware Control to Set the Maximum Number of Rows Processed to Render a Table.....	7-6
7.4	About the Oracle BI Server Query Cache.....	7-7
7.4.1	Query Cache Architecture	7-8
7.4.2	Advantages of Caching.....	7-8
7.4.3	Costs of Caching	7-9
7.4.3.1	Disk Space.....	7-9
7.4.3.2	Administrative Tasks	7-9
7.4.3.3	Keeping the Cache Up To Date	7-10
7.4.3.4	CPU Usage and Disk I/O.....	7-10
7.4.4	Cache Sharing Across Users.....	7-10
7.4.5	About the Refresh Interval for XML Data Sources	7-10
7.4.6	About the Global Cache.....	7-11
7.5	Configuring Query Caching.....	7-12
7.5.1	Using Fusion Middleware Control to Enable and Disable Query Caching.....	7-13
7.5.2	Using Fusion Middleware Control to Set Query Cache Parameters.....	7-13
7.5.3	Manually Editing Additional Query Cache Parameters.....	7-14
7.5.4	Using Fusion Middleware Control to Set Global Cache Parameters	7-14
7.5.5	Manually Editing Additional Global Cache Parameters	7-15
7.6	Monitoring and Managing the Cache	7-15
7.6.1	Choosing a Cache Management Strategy	7-15
7.6.1.1	Disable Caching for the System.....	7-15
7.6.1.2	Caching and Cache Persistence Timing for Specified Physical Tables.....	7-16
7.6.1.3	Configure Oracle BI Server Event Polling Tables	7-16
7.6.2	Purging and Maintaining Cache Using ODBC Procedures.....	7-16
7.6.2.1	About ODBC Procedure Syntax	7-17
7.6.2.2	About Sharing the Presentation Services Query Cache.....	7-18
7.6.2.3	About Result Records	7-18
7.6.2.4	Storing and Purging Cache for SAP/BW Data Sources.....	7-18
7.6.3	How Repository Changes Affect the Query Cache	7-20
7.6.3.1	Online Mode.....	7-20
7.6.3.2	Offline Mode	7-20

7.6.3.3	Switching Between Repositories	7-20
7.6.3.4	Changes to Dynamic Repository Variables	7-20
7.7	Strategies for Using the Cache	7-21
7.7.1	About Cache Hits.....	7-21
7.7.1.1	Ensuring Correct Cache Results When Using Row-Level Database Security ..	7-23
7.7.2	Running a Suite of Queries to Populate the Cache.....	7-24
7.7.3	Using Agents to Seed the Oracle BI Server Cache	7-25
7.7.4	Using the Cache Manager.....	7-25
7.7.4.1	Displaying Global Cache Information in the Cache Manager	7-26
7.7.4.2	Purging Cache in the Administration Tool.....	7-27
7.8	Cache Event Processing with an Event Polling Table	7-28
7.8.1	Setting Up Event Polling Tables on the Physical Databases	7-29
7.8.1.1	Polling Table Structure	7-29
7.8.1.2	Sample Event Polling Table CREATE TABLE Statements	7-31
7.8.2	Making the Event Polling Table Active.....	7-32
7.8.3	Populating the Oracle BI Server Event Polling Table.....	7-32
7.8.4	Troubleshooting Problems with Event Polling Tables.....	7-33
7.9	Managing the Oracle BI Presentation Services Cache Settings	7-33
7.10	Improving Oracle BI Web Client Performance.....	7-35
7.10.1	Configuring Apache HTTP Server for Static File Caching	7-35
7.10.2	Configuring Oracle HTTP Server for Static File Caching	7-37

Part IV Resolving Issues

8 Diagnosing and Resolving Issues in Oracle Business Intelligence

8.1	How Are Issues Resolved?	8-1
8.2	Viewing And Configuring Diagnostic Log Files.....	8-2
8.2.1	Using Fusion Middleware Control to View Log Information, Error Messages, and Alerts	8-2
8.2.2	Configuring Log File Rotation Policy and Specifying Log Levels	8-3
8.2.2.1	Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels.....	8-3
8.2.2.2	Manually Changing Additional Log File Settings.....	8-4
8.3	Understanding Diagnostic Log and Log Configuration Files	8-5
8.3.1	What Are Diagnostic Log Files and Where Are They Located?	8-5
8.3.2	What Are Diagnostic Log Configuration Files and Where Are They Located?	8-6
8.3.3	What Are Log File Message Categories and Levels?.....	8-8
8.3.4	What is Log File Rotation?.....	8-9
8.3.5	What Messages Are Included in the System Log?.....	8-10
8.4	Managing the Query Log.....	8-10
8.4.1	Configuring Query Logging	8-11
8.4.1.1	Setting the Query Logging Level	8-11
8.4.1.2	Setting the Query Logging Level for a User	8-13
8.4.2	Using the Log Viewer.....	8-13
8.4.2.1	Running the Log Viewer Utility	8-13
8.4.2.2	Interpreting the Log Records.....	8-14
8.5	Logging in Oracle BI Presentation Services	8-14

8.5.1	Using the Oracle BI Presentation Services Logging Facility	8-15
8.5.2	Structure for the Oracle BI Presentation Services Configuration File.....	8-15
8.5.3	Examples of the Formats of Logged Messages.....	8-19
8.5.4	Oracle BI Presentation Services Message Structure.....	8-20
8.5.5	Oracle BI Presentation Services Log Filters	8-20
8.5.6	Diagnosing Issues with Agents	8-21

9 Managing Usage Tracking

9.1	Setting Up Usage Tracking	9-1
9.1.1	Setting Up Direct Insertion to Collect Information for Usage Tracking.....	9-2
9.1.2	Setting Up a Log File to Collect Information for Usage Tracking	9-3
9.1.2.1	Selecting an Output Location	9-3
9.1.2.2	File Naming Conventions	9-4
9.1.2.3	Output File Format.....	9-4
9.1.2.4	Performance Considerations.....	9-6
9.2	Description of the Usage Tracking Data.....	9-6

Part V Configuring Oracle Business Intelligence

10 Configuring Repositories

10.1	Why Upload a Repository and Set the Catalog Location?.....	10-1
10.1.1	Understanding Modes	10-1
10.1.2	Understanding Repository Version Numbers.....	10-2
10.2	Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location	10-2
10.3	Using the System MBeans to Upload the Metadata Repository	10-3

11 Configuring Connections to External Systems

11.1	Configuring E-Mail and Agents	11-1
11.1.1	Using Fusion Middleware Control to Configure Oracle BI Scheduler E-Mail Settings that Affect Agents.....	11-1
11.2	Configuring for Actions with the Action Framework.....	11-2
11.3	Configuring Connections to Data Sources	11-2
11.4	Configuring for Connections to the Marketing Content Server	11-3
11.4.1	Using Fusion Middleware Control to Configure the Connection to the Marketing Content Server.....	11-3
11.4.2	Manually Editing Settings for the Marketing Content Server Connection.....	11-4
11.5	Configuring to Download Oracle Hyperion Smart View for Office	11-4

12 Configuring Presentation Setting Defaults

12.1	Using Fusion Middleware Control to Change Presentation Setting Defaults	12-1
------	---	------

13 Configuring Mapping and Spatial Information

13.1	What Are the System Requirements for Map Views?	13-1
13.2	Hardware Sizing and Deployment Strategy for Maps.....	13-3
13.3	Administering Maps.....	13-3

13.3.1	Working with Maps and Layers.....	13-3
13.3.1.1	Associating Layers with Columns	13-3
13.3.1.2	Ordering Layers on Maps	13-4
13.3.1.3	Changes to Spatial Metadata Require Restart	13-6
13.3.2	Administration Page Functions	13-6
13.3.3	Administering Maps Using Administration Pages	13-7
13.3.4	Handling the Translation of Layers in Maps.....	13-8

14 Configuring Time Zones

14.1	Why and Where are Time Zones Used?	14-1
14.2	Setting Time Zones	14-2
14.3	What is the Precedence Order for Time Zones.....	14-3
14.3.1	User-Preferred Time Zone.....	14-3
14.4	Where are Time Zone Specifications Stored?	14-3
14.4.1	Specifying Time Zone Values	14-3
14.5	Description of Time Zone Settings	14-4
14.6	Example: Configuration File Settings for Specifying the Time Zone	14-5

15 Localizing Oracle Business Intelligence

15.1	What Is Localization?	15-1
15.1.1	What Components Are Translated?.....	15-1
15.1.2	Tasks for Localizing Oracle Business Intelligence Components	15-2
15.2	Localizing Oracle BI Presentation Services	15-3
15.2.1	Localizing the User Interface for Oracle BI Presentation Services	15-3
15.2.1.1	Understanding the Directory Structure for Localizing Presentation Services .	15-3
15.2.1.2	Localizing Messages for Users' Preferred Currency	15-4
15.2.1.3	Specifying the Default Language for the Sign-In Page	15-4
15.2.1.4	Configuring the Languages and Locales for the Sign-In Page.....	15-5
15.2.1.5	Specifying the Language in the URL	15-6
15.2.2	Localizing Oracle BI Presentation Catalog Captions.....	15-7
15.2.2.1	Step 1: Understanding the Export Process.....	15-7
15.2.2.2	Step 2: Exporting Text Strings in the Catalog.....	15-7
15.2.2.3	Step 3: Editing Exported Strings in XML Files	15-8
15.2.2.4	Step 4: Handling Duplicate Exported Text Strings.....	15-9
15.2.2.5	Step 5: Exposing Text Strings in the Catalog.....	15-11
15.2.3	Tip for Arabic and Hebrew in Mozilla Firefox Browsers	15-12
15.3	Setting the Current Locale in Catalog Manager	15-12
15.4	Setting the Current Locale in the Oracle BI Server	15-13
15.4.1	Setting Locale Parameters on the Oracle BI Server.....	15-13
15.4.1.1	Setting the Locale on UNIX Systems	15-14
15.4.2	Understanding How the Error Message Language Is Determined.....	15-15
15.4.3	Troubleshooting the Current Locale in the Oracle BI Server	15-15
15.4.3.1	Handling the NLS Locale Not Supported Error Message	15-16
15.4.3.2	Setting the Japanese Locale on AIX Systems	15-16
15.5	Localizing Metadata Names in the Repository	15-16
15.6	Supporting Multilingual Data.....	15-18

15.6.1	What is Multilingual Data Support?	15-18
15.6.2	What is Lookup?	15-19
15.6.3	What is Double Column Support?	15-19
15.6.4	Designing Translation Lookup Tables in a Multilingual Schema	15-19
15.6.4.1	A Lookup Table for Each Base Table	15-19
15.6.4.2	A Lookup Table for Each Translated Field	15-20
15.6.5	Creating Logical Lookup Tables and Logical Lookup Columns	15-20
15.6.5.1	Creating Logical Lookup Tables	15-20
15.6.5.2	Designating a Logical Table as a Lookup Table.....	15-22
15.6.5.3	About the LOOKUP Function Syntax	15-22
15.6.5.4	Creating Logical Lookup Columns.....	15-23
15.6.6	Creating Physical Lookup Tables and Physical Lookup Columns	15-24
15.6.7	Supporting Multilingual Data in Essbase Through Alias Tables	15-26
15.6.8	Enabling Lexicographical Sorting	15-26

16 Configuring Currency Options

16.1	Changing the Default Currency for Analyses	16-1
16.2	Defining User-Preferred Currency Options.....	16-2
16.2.1	Defining User-Preferred Currency Options Using a Static Mapping	16-3
16.2.2	Example: Static Mapping to Define User-Preferred Currency Options	16-4
16.2.3	Defining User-Preferred Currency Options Using a Dynamic Mapping.....	16-5
16.2.4	Example: Dynamic Mapping to Define User-Preferred Currency Options.....	16-6

17 Configuring and Managing the Oracle BI Presentation Catalog

17.1	About the Oracle BI Presentation Catalog	17-1
17.1.1	Objects in the Catalog.....	17-1
17.1.1.1	Guidelines for Object Names	17-2
17.1.1.2	Attribute Files for Objects.....	17-3
17.1.1.3	Lock Files for Objects	17-3
17.1.2	Locations for the Catalog.....	17-3
17.1.3	File System Guidelines for Catalogs	17-4
17.1.3.1	Handling Users of the Catalog	17-4
17.1.3.2	Handling Heterogeneous Nodes.....	17-4
17.1.3.3	Handling Catalog Files on Various Platforms.....	17-5
17.1.3.4	Known Issues with Catalog Files	17-5
17.2	Maintaining the Oracle BI Presentation Catalog	17-6
17.2.1	Manually Changing Additional Configuration Settings for the Catalog	17-6
17.2.2	Manually Creating a New Oracle BI Presentation Catalog	17-7
17.2.3	Deploying Catalogs and Objects to Production	17-7
17.2.3.1	Deploying Catalogs to Production.....	17-7
17.2.3.2	Deploying Objects to Production	17-8
17.2.4	Validating the Catalog	17-9
17.2.4.1	Performing a Basic Validation of the Catalog	17-9
17.2.4.2	Specifying the Elements for Validating the Catalog.....	17-10
17.3	About Catalog Manager	17-11
17.3.1	Uses for Catalog Manager	17-12
17.3.2	Guidelines for Working with Catalog Manager.....	17-12

17.3.3	Tips for Working with Catalog Manager	17-12
17.4	Starting Catalog Manager and Opening Catalogs	17-13
17.4.1	Starting Catalog Manager.....	17-13
17.4.2	Understanding the Two Catalog Modes	17-13
17.4.2.1	Online Mode.....	17-13
17.4.2.2	Offline Mode	17-14
17.4.3	Operations Available in Online Mode and Offline Mode	17-14
17.4.4	Opening an Oracle BI Presentation Catalog	17-15
17.5	Using the Catalog Manager Workspace	17-16
17.5.1	What Does the Catalog Manager Workspace Do?.....	17-16
17.5.2	What Does the Catalog Manager Workspace Look Like?	17-16
17.5.3	Managing the View of the Catalog Manager Workspace.....	17-17
17.6	Working with Objects in Catalog Manager.....	17-17
17.6.1	Searching for Catalog Objects Using Catalog Manager.....	17-18
17.6.2	Copying and Pasting Objects	17-19
17.6.2.1	Tips for Copying and Pasting	17-19
17.6.2.1.1	Copying and Pasting Objects Between Catalogs Using Menus	17-19
17.6.2.2	Advanced Options for Pasting Objects	17-19
17.6.2.2.1	Paste Overwrite	17-20
17.6.2.2.2	Paste ACL	17-20
17.6.3	Renaming Catalog Objects	17-22
17.6.4	Working with the Properties of Catalog Objects.....	17-22
17.6.5	Setting Permissions of Catalog Objects	17-23
17.6.6	Previewing Objects from Catalog Manager.....	17-25
17.6.7	Working with Tenants	17-25
17.6.7.1	What Happens When Tenants are Provisioned?	17-25
17.6.7.2	What Options are Available for Provisioning?	17-25
17.6.7.3	How is Multi-Tenancy Enabled?.....	17-26
17.7	Viewing and Editing Catalog Objects in XML.....	17-26
17.8	Searching for and Replacing Catalog Text Using Catalog Manager.....	17-28
17.8.1	Searching for and Replacing a Simple Catalog Text String.....	17-28
17.8.2	About Searching for and Replacing Multiple Catalog Text Strings.....	17-28
17.8.2.1	XML File Format for Searching for and Replacing Text Strings.....	17-28
17.8.2.2	Example XML File for Searching for and Replacing Text Strings.....	17-29
17.8.3	Searching for and Replacing Multiple Catalog Text Strings	17-29
17.9	Creating Reports to Display Catalog Data Using Catalog Manager.....	17-30
17.10	Archiving and Unarchiving Using Catalog Manager	17-31
17.10.1	Archiving a Folder Using Catalog Manager.....	17-31
17.10.2	Unarchiving a Folder Using Catalog Manager	17-32
17.11	Configuring for Full-Text Catalog Search	17-32
17.11.1	What is a Full-Text Catalog Search?.....	17-32
17.11.2	Deployment of Oracle Full-Text Catalog Search.....	17-33
17.11.3	How is Security Handled in the Full-Text Catalog Search?	17-33
17.11.4	Process for Configuring the Full-Text Catalog Search	17-34
17.11.4.1	Step 1: Assigning Privileges for Crawling and Indexing.....	17-34
17.11.4.2	Step 2: Ensuring that Secure Enterprise Search is Available.....	17-34
17.11.4.3	Step 3: Configuring Identity Management for Full-Text Catalog Search.....	17-35

17.11.4.4	Step 4: Creating the Data Source for Full-Text Catalog Search.....	17-35
17.11.4.5	Step 5: Editing the Schedule for Full-Text Catalog Search	17-37
17.11.4.6	Step 6: Specifying the Look and Feel of Search Results	17-37
17.11.4.7	Step 7: Enabling Incremental Crawls for Full-Text Catalog Search	17-38
17.11.4.8	Step 8: Configuring Properties for Full-Text Catalog Search Manually	17-38
17.11.4.9	Step 9: Allowing Users to Access the Full-Text Catalog Search	17-39
17.11.4.9.1	Providing Designers with the URL.....	17-39
17.11.4.9.2	Making the Shortcut Available from the Oracle BI EE Home Page.....	17-39
17.12	Replicating Oracle BI Presentation Catalogs	17-40
17.12.1	About Replication Log Files	17-40
17.12.1.1	Replication Log File Records	17-41
17.12.2	Configuring to Replicate an Oracle BI Presentation Catalog	17-42
17.12.3	Copying Oracle BI Presentation Catalog Files.....	17-42
17.12.4	Using the Oracle BI Presentation Services Replication Agent	17-42
17.12.4.1	mark.....	17-43
17.12.4.2	remotecopy	17-43
17.12.4.3	run.....	17-44
17.12.5	Creating the config.xml File for Replication	17-44
17.12.5.1	Structure of the Configuration File	17-44
17.12.5.2	Elements of the Configuration File	17-44
17.12.5.3	Example config.xml File	17-47
17.12.6	Editing the instanceconfig.xml File for Replication.....	17-47
17.12.7	Resuming Replication if Replication Log Files Are Unavailable.....	17-49

Part VI Advanced Configuration Settings

18 Configuring and Managing Analyses and Dashboards

18.1	Managing Dashboards	18-1
18.2	Performing General Configuration Tasks for Analyses	18-2
18.2.1	Providing Access to Metadata Dictionary Information	18-2
18.2.2	Supporting Nested Folders, Navigation, and Drill-Down	18-3
18.3	Configuring for Displaying and Processing Data in Views	18-3
18.3.1	Manually Configuring for Data in Views	18-4
18.3.1.1	Manually Configuring Cube Settings for Pivot Tables and Graphs	18-4
18.3.1.2	Manually Configuring Settings for Data in Views	18-4
18.3.2	Manually Configuring for Graphs and Gauges	18-6
18.3.2.1	Configuring Fonts for Graphs	18-8
18.3.2.1.1	Setting the Embed Fonts Element	18-8
18.3.2.1.2	Delivering Font Files for Printing	18-8
18.3.3	Manually Changing Alternating Bar Color	18-9
18.4	Manually Changing Presentation Settings.....	18-9
18.4.1	Manually Changing Presentation Setting Defaults	18-10
18.4.2	Configuring Links to Dashboard Pages	18-11
18.4.3	Configuring an Alternate Toolbar for Oracle BI Publisher	18-12
18.4.4	Modifying the Table of Contents for PDF Versions of Briefing Books	18-13
18.5	Blocking Analyses in Answers.....	18-13
18.5.1	Storing JavaScript Files	18-14

18.5.2	Blocking Analyses Based on Criteria	18-14
18.5.3	Blocking Analyses Based on Formula.....	18-15
18.5.4	Validation Helper Functions	18-16
18.6	Specifying View Defaults for Analyses and Dashboards	18-17
18.6.1	XML Message Files for View Defaults.....	18-17
18.6.2	Examples of Customizing Default Values for Analyses and Dashboards	18-17
18.6.2.1	Adding a Default Header or Footer to New Analyses.....	18-18
18.6.2.2	Preventing Auto-Previewing of Results.....	18-18
18.6.2.3	Setting Defaults for Analyses in the Compound Layout.....	18-19
18.6.2.4	Changing Dashboards Section Defaults.....	18-19
18.6.2.5	Specifying Dashboard Page Defaults Including Headers and Footers	18-19
18.7	Configuring for Write Back in Analyses and Dashboards.....	18-20
18.7.1	Write-Back Limitations	18-20
18.7.2	Configuring for Write Back	18-21
18.7.2.1	Setting the LightWriteback Element	18-22
18.7.3	About the Write-Back Template	18-22
18.7.3.1	How Write Back Works	18-22
18.7.3.2	Requirements for the Write-Back Template	18-23
18.7.3.3	Example: Write-Back Template	18-24

19 Configuring and Managing Agents

19.1	How Are Agents Used?	19-1
19.2	How Do Antivirus Software and Privileges Affect Agents?	19-2
19.2.1	How Does Antivirus Software Affect Agents?.....	19-2
19.2.2	What Privileges Affect Agents?	19-2
19.3	Configuring Settings That Affect Agents	19-2
19.3.1	Manually Configuring Presentation Services Settings That Affect Agents	19-3
19.3.2	Manually Changing Additional Scheduler Settings That Affect Agents	19-4
19.3.3	What Additional Scheduler Configuration Settings Affect Agents?	19-5
19.3.3.1	General Scheduler Configuration Settings That Affect Agents	19-5
19.3.3.2	E-Mail Scheduler Configuration Settings That Affect Agents	19-7
19.3.3.3	Agent Scheduler Configuration Settings	19-8
19.3.4	Controlling Delivery Options for Agents.....	19-9
19.4	Managing Device Types for Agents	19-9
19.5	Monitoring Active Agent Sessions	19-10

20 Configuring Advanced Options for Mapping and Spatial Information

20.1	Configuring MapViewer to Support Map Views.....	20-1
20.2	Manually Configuring for Map Views	20-2
20.3	Inserting Text on a Map	20-4
20.4	Configuring Maps for External Consumption.....	20-4

21 Configuring Resource Availability and URL Generation

Part VII Automating Management of Oracle Business Intelligence

22 Introducing the Oracle BI Systems Management API

22.1	What Is the Oracle BI Systems Management API?	22-1
22.2	Accessing Oracle BI Systems Management API Methods	22-2
22.3	Using the Oracle BI Systems Management API to Manage and Configure Oracle Business Intelligence	22-2
22.3.1	Displaying Oracle Business Intelligence Status Using the Oracle BI Systems Management API	22-2
22.3.1.1	Example: Using WLST Commands in a Script to Display the Status of Oracle Business Intelligence	22-3
22.3.1.2	Example: Using a Java Program to Display the Status of Oracle Business Intelligence.....	22-4
22.3.2	Starting and Stopping Oracle Business Intelligence Using the Oracle BI Systems Management API	22-7
22.3.2.1	Example: Using WLST Commands to Stop and Start Oracle Business Intelligence.....	22-7
22.3.2.2	Example: Using a Java Program to Stop and Start Oracle Business Intelligence.....	22-9
22.3.3	Updating Configuration Settings Using the Oracle BI Systems Management API	22-9
22.3.3.1	Example: Using WLST to Edit the E-Mail Sender Display Name Configuration Setting	22-9
22.3.3.2	Example: Using a Java Program to Edit the E-Mail Sender Display Name Configuration Setting.....	22-12
22.3.4	Scaling Out for High Availability and Performance Using the Oracle BI Systems Management API	22-14
22.3.4.1	Example: Using WLST to Scale Out to Servers	22-14
22.3.4.2	Example: Using a Java Program to Scale Out to Servers	22-18
22.3.5	Capturing Metrics Using the Oracle BI Systems Management API.....	22-21
22.3.5.1	Using the Dynamic Monitoring Service for Metrics.....	22-21
22.3.5.2	Using WLTS Commands for Metrics.....	22-22
22.4	Descriptions of JMX Admin MBeans and Methods for the Oracle BI Systems Management API	22-25

Part VIII Managing the Life Cycle

23 Patching Oracle Business Intelligence Systems

23.1	What Is Patched for the Oracle Business Intelligence Platform?	23-1
23.2	Patching the Platform	23-2
23.2.1	What Conditions Apply when Patching the Platform?	23-2
23.2.2	Applying a Patch to the Platform	23-2
23.2.2.1	Step 1: Download the Patch	23-2
23.2.2.2	Step 2: Stop System Component Processes for Oracle BI EE	23-3
23.2.2.3	Step 3: Apply the Patch to Each Oracle Home	23-3
23.2.2.4	Step 4: Start System Component Processes for Oracle BI EE	23-3
23.2.2.5	Step 5: Redeploy Patched J2EE Applications	23-3
23.2.3	What Happens If a Patching Conflict Occurs?	23-4
23.3	Rolling Back a Platform Patch.....	23-4
23.3.1	Step 1: Stop System Component Processes for Oracle BI EE.....	23-5
23.3.2	Step 2: Invoke OPatch with "-rollback" Option	23-5

23.3.3	Step 3: Start System Component Processes for Oracle BI EE	23-5
23.3.4	Step 4: Redeploy J2EE Applications.....	23-5
23.4	Determining Current Patch Levels.....	23-6

24 Moving Between Environments

25 Backup and Recovery of Oracle Business Intelligence Systems

Part IX Reference Information

A NQSCONFIG.INI File Configuration Settings

A.1	About Parameters in the NQSCONFIG.INI File	A-1
A.1.1	How to Update Parameters in NQSCONFIG.INI.....	A-2
A.2	Repository Section Parameters	A-3
A.3	Query Result Cache Section Parameters	A-3
A.3.1	ENABLE	A-4
A.3.2	DATA_STORAGE_PATHS	A-4
A.3.3	MAX_ROWS_PER_CACHE_ENTRY	A-5
A.3.4	MAX_CACHE_ENTRY_SIZE	A-6
A.3.5	MAX_CACHE_ENTRIES	A-6
A.3.6	POPULATE_AGGREGATE_ROLLUP_HITS.....	A-6
A.3.7	USE_ADVANCED_HIT_DETECTION	A-7
A.3.7.1	Reasons Why a Query Is Not Added to the Cache.....	A-7
A.3.8	MAX_SUBEXPR_SEARCH_DEPTH.....	A-8
A.3.9	DISABLE_SUBREQUEST_CACHING	A-8
A.3.10	GLOBAL_CACHE_STORAGE_PATH.....	A-8
A.3.11	MAX_GLOBAL_CACHE_ENTRIES.....	A-9
A.3.12	CACHE_POLL_SECONDS	A-9
A.3.13	CLUSTER_AWARE_CACHE_LOGGING.....	A-9
A.4	General Section Parameters.....	A-9
A.4.1	LOCALE.....	A-9
A.4.2	SORT_ORDER_LOCALE	A-10
A.4.2.1	SORT_ORDER_LOCALE on UNIX Operating Systems.....	A-11
A.4.3	SORT_TYPE.....	A-11
A.4.4	CASE_SENSITIVE_CHARACTER_COMPARISON	A-11
A.4.5	NULL_VALUES_SORT_FIRST.....	A-13
A.4.6	DATE_TIME_DISPLAY_FORMAT.....	A-13
A.4.7	DATE_DISPLAY_FORMAT.....	A-13
A.4.8	TIME_DISPLAY_FORMAT.....	A-13
A.4.9	WORK_DIRECTORY_PATHS.....	A-13
A.4.10	VIRTUAL_TABLE_PAGE_SIZE.....	A-14
A.4.11	USE_LONG_MONTH_NAMES.....	A-15
A.4.12	USE_LONG_DAY_NAMES.....	A-15
A.4.13	UPPERCASE_USERNAME_FOR_INITBLOCK	A-15
A.5	Security Section Parameters	A-15
A.5.1	DEFAULT_PRIVILEGES	A-15

A.5.2	PROJECT_INACCESSIBLE_COLUMN_AS_NULL	A-15
A.5.3	IGNORE_LDAP_PWD_EXPIRY_WARNING.....	A-15
A.5.4	SSL.....	A-16
A.5.5	SSL_CERTIFICATE_FILE	A-16
A.5.6	SSL_PRIVATE_KEY_FILE.....	A-16
A.5.7	SSL_VERIFY_PEER	A-17
A.5.8	SSL_CA_CERTIFICATE_FILE	A-17
A.5.9	SSL_TRUSTED_PEER_DNS.....	A-17
A.5.10	SSL_CERT_VERIFICATION_DEPTH	A-17
A.5.11	SSL_CIPHER_LIST	A-17
A.6	Server Section Parameters	A-17
A.6.1	READ_ONLY_MODE.....	A-18
A.6.2	MAX_SESSION_LIMIT.....	A-18
A.6.2.1	About the MAX_SESSION_LIMIT and SERVER_THREAD_RANGE Parameters	A-18
A.6.3	MAX_REQUEST_PER_SESSION_LIMIT.....	A-19
A.6.4	SERVER_THREAD_RANGE	A-19
A.6.5	SERVER_THREAD_STACK_SIZE	A-19
A.6.6	DB_GATEWAY_THREAD_RANGE	A-20
A.6.7	DB_GATEWAY_THREAD_STACK_SIZE.....	A-20
A.6.8	MAX_EXPANDED_SUBQUERY_PREDICATES	A-20
A.6.9	MAX_QUERY_PLAN_CACHE_ENTRIES.....	A-20
A.6.10	MAX_DRILLDOWN_INFO_CACHE_ENTRIES.....	A-20
A.6.11	MAX_DRILLDOWN_QUERY_CACHE_ENTRIES.....	A-21
A.6.12	INIT_BLOCK_CACHE_ENTRIES.....	A-21
A.6.13	CLIENT_MGMT_THREADS_MAX.....	A-21
A.6.14	RPC_SERVICE_OR_PORT	A-21
A.6.15	ENABLE_DB_HINTS.....	A-22
A.6.16	PREVENT_DIVIDE_BY_ZERO	A-22
A.6.17	CLUSTER_PARTICIPANT.....	A-22
A.6.18	REPOSITORY_PUBLISHING_DIRECTORY	A-23
A.6.19	REQUIRE_PUBLISHING_DIRECTORY	A-24
A.6.20	DISCONNECTED.....	A-24
A.6.21	AUTOMATIC_RESTART.....	A-24
A.6.22	VARIABLE_VALUE_LIMIT	A-24
A.6.23	FMW_SECURITY_SERVICE_URL.....	A-25
A.6.24	FMW_SECURITY_SERVICE_MAX_NUMBER_OF_CONNECTIONS.....	A-25
A.6.25	FMW_SECURITY_SERVICE_MAX_NUMBER_OF_RETRIES	A-25
A.6.26	FMW_UPDATE_ROLE_AND_USER_REF_GUIDS	A-25
A.7	Dynamic Library Section Parameters	A-25
A.8	Usage Tracking Section Parameters	A-26
A.8.1	ENABLE	A-27
A.8.2	DIRECT_INSERT	A-27
A.8.3	STORAGE_DIRECTORY	A-28
A.8.4	CHECKPOINT_INTERVAL_MINUTES.....	A-28
A.8.5	FILE_ROLLOVER_INTERVAL_MINUTES.....	A-28
A.8.6	CODE_PAGE.....	A-29
A.8.7	PHYSICAL_TABLE_NAME	A-29

A.8.8	CONNECTION_POOL	A-30
A.8.9	BUFFER_SIZE.....	A-30
A.8.10	BUFFER_TIME_LIMIT_SECONDS.....	A-30
A.8.11	NUM_INSERT_THREADS	A-30
A.8.12	MAX_INSERTS_PER_TRANSACTION.....	A-30
A.8.13	SUMMARY_STATISTICS_LOGGING	A-30
A.8.14	SUMMARY_ADVISOR_TABLE_NAME	A-31
A.9	Query Optimization Flags Section Parameters	A-31
A.9.1	STRONG_DATETIME_TYPE_CHECKING	A-31
A.10	MDX Member Name Cache Section Parameters.....	A-31
A.10.1	ENABLE	A-31
A.10.2	DATA_STORAGE_PATH	A-31
A.10.3	MAX_SIZE_PER_USER	A-31
A.10.4	MAX_MEMBER_PER_LEVEL.....	A-31
A.10.5	MAX_CACHE_SIZE.....	A-32
A.11	Aggregate Persistence Section Parameters.....	A-32
A.11.1	AGGREGATE_PREFIX	A-32
A.11.2	AGGREGATE_THREAD_POOL_SIZE	A-32
A.11.3	AGGREGATE_AW_NAME.....	A-32
A.11.4	PREAGGREGATE_AW_CUBE	A-32
A.12	JavaHost Section Parameters.....	A-32
A.12.1	JAVAHOST_HOSTNAME_OR_IP_ADDRESSES	A-33
A.13	Datamart Automation Section Parameters	A-33
A.13.1	ESSBASE_STUDIO_URL	A-33

B Advanced Configuration Reference

B.1	Making Advanced Configuration Changes for Presentation Services	B-1
B.1.1	Protecting Pages in Oracle BI EE from Attack.....	B-2
B.2	Using the JavaHost Service for Oracle BI Presentation Services.....	B-3

C Mapping User Interface Labels with Configuration File Elements

D Manually Updating Fusion Middleware Control Settings

D.1	Disabling Fusion Middleware Control from Managing Oracle Business Intelligence Configuration Settings	D-1
D.2	Manually Updating Oracle Business Intelligence Configuration Settings Normally Managed by Fusion Middleware Control	D-2

Glossary

Index

Preface

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

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

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

This guide contains 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

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

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 11g Release 1 (11.1.1) documentation set:

- The Oracle Business Intelligence chapter in the *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.
<i>italic</i>	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 preface describes changes to system administration features for Oracle Business Intelligence Enterprise Edition 11g 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 for Oracle BI EE 11g Release 1 \(11.1.1.5\)"](#)
- ["New Features for Oracle BI EE 11g Release 1 \(11.1.1.3\)"](#)

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

This section contains the following topic:

- ["New Features"](#)

New Features

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

- The Simple Install type has been changed. In this release, only an Administration server is configured as part of this type. The Managed Server and the Node Manager are not installed, which affects the way that certain administration tasks are performed. See "Installation Types" in *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence* for more information.
- The process of exporting captions from the catalog for translation has been enhanced to address duplicate captions. For information, see [Section 15.2.2.4, "Step 4: Handling Duplicate Exported Text Strings."](#)
- You can use Catalog Manager to provision tenants. For information, see [Section 17.6.7, "Working with Tenants."](#)

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

This section contains the following topics:

- ["New Features"](#)
- ["Upgrade Considerations"](#)

New Features

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

- ["Integrated Management Experience"](#)
- ["Centralized System Administration"](#)
- ["Programmatic and Scripting Capabilities for Administrative Operations"](#)
- ["New Configuration Settings for Views"](#)
- ["Validation of the Catalog"](#)
- ["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 must 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 22, "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 about configuring these views, see [Chapter 18, "Configuring and Managing Analyses and Dashboards."](#)

Validation of the Catalog

This release provides functionality that enables you to validate the catalog to address inconsistencies that have developed over time., For information, see [Section 17.2.4.1, "Performing a Basic Validation of the Catalog."](#)

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 "Working with Catalog Groups" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*

Enhancements for Localization

This release introduces several enhancements for localizing your system, including lookup tables and alias tables. For information, see [Chapter 15, "Localizing Oracle Business Intelligence."](#)

Upgrade Considerations

The following list identifies what you must be aware of if your site is upgrading to Oracle BI EE 11g Release 1 (11.1.1.3):

- The Everyone group has been replaced with the AuthenticatedUser role. For information, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

For more information about upgrading to Oracle BI EE 11g, 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 includes 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"](#)

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 Oracle Business Intelligence architecture; lists the tools that can help you complete system administration tasks, and provides links to system requirements and certification information.

This chapter includes 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, "What System Administration Tools Manage Oracle Business Intelligence?"](#)
- [Section 1.5, "Working with the Sample Application"](#)
- [Section 1.6, "Oracle BI Publisher Integration"](#)
- [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 Sockets 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 scaling the environment, deploying for high availability, performance tuning, and query caching.
Diagnosing problems and resolving issues	Part IV, "Resolving Issues" Contains chapters about diagnosing and resolving issues and about usage tracking.

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

System Administration Task	More Information
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	For more information, see <i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i>
Automating management of an Oracle Business Intelligence system	Part VII, "Automating Management of Oracle Business Intelligence" Describes the Oracle BI Systems Management API.
Managing the life cycle.	Part VIII, "Managing the Life Cycle" Contains chapters about life cycle management tasks such as patching, moving between environments, and backup and recovery.
Securing the system	<ul style="list-style-type: none"> ■ Defines administrative role membership <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i> ■ Secures middle-tier communications Secure Sockets Layer (SSL) and Single Sign-On (SSO) are not described in this guide. For information, see <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>.

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.

Note: You can improve the performance of your production system by using a Web server with Oracle Business Intelligence, such as Oracle HTTP Server or Apache HTTP Server. A Web server is not included by default in the Oracle Business Intelligence installer and is not part of the Oracle Business Intelligence system logical architecture. You must install and configure a Web server separately.

1.3.1 Oracle Business Intelligence on a Single Host

You can install Oracle Business Intelligence on a single host using either an Enterprise Install type or a Simple Install type. The architecture differs slightly depending on install type, as described in the following sections.

See "Installation Types" in *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence* for more information.

1.3.1.1 A Single Host and Enterprise Install

In an Enterprise Install type, the following components are installed in the Oracle BI Domain on the single host. The BI Domain consists of Java components that are deployed into one or more Java EE (JEE) containers within a single WebLogic server domain; system (non-JEE) components and processes; and required configuration files, metadata repositories, and infrastructure.

- **Java Components** — Deployed as JEE applications to service SOAP, HTTP, and other forms of requests.

For more information, see [Section 1.3.3, "About Java Components and System Components for Oracle Business Intelligence."](#)
- **Administration Server** — Deployed as a JEE container that runs in a dedicated Java virtual machine that contains Java components for administering the system. These components include Oracle WebLogic Server Administration Console, Oracle Fusion Middleware Control, and JMX MBeans.

For more information, see [Section 1.3.4, "What Is the Administration Server?"](#)
- **Managed Server** — Deployed as 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. These services and applications include BI Publisher and Oracle Real-Time Decisions.

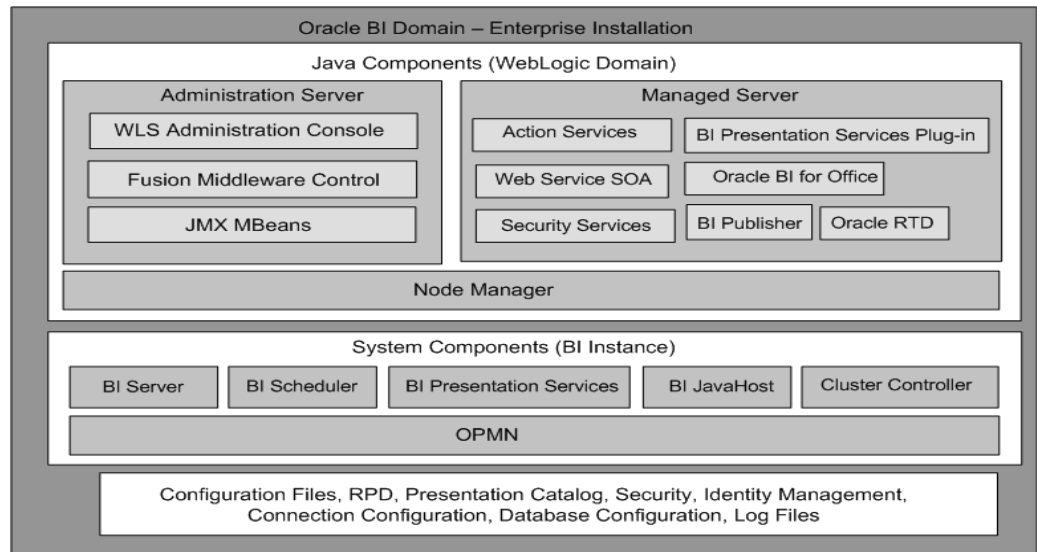
An Oracle BI domain contains one or more Managed Servers that are distributed across one or more host computers.
- **Node Manager** — Provides process management services for the Administration Server and Managed Server processes.

For more information, see *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server*.
- **System Components** — Deployed as server processes and provide the core services that enable Oracle Business Intelligence, including OPMN. OPMN (Oracle Process Manager and Notification server) maintains the Oracle Business Intelligence system component processes.

For more information, see [Section 1.3.3, "About Java Components and System Components for Oracle Business Intelligence."](#) For more information about OPMN, see [Section 1.4.3, "Oracle Process Manager and Notification Server \(OPMN\) Tool."](#)
- **Other Domain Contents** — Includes all the necessary software, metadata, configuration files, RPD files, Oracle BI Presentation Catalog, and connection and database configuration information that are required to run an Oracle Business Intelligence system.

[Figure 1-1](#) illustrates the Oracle Business Intelligence system architecture on a single host for an Enterprise Install type.

Figure 1–1 System Logical Architecture for Enterprise Install (Single Host)



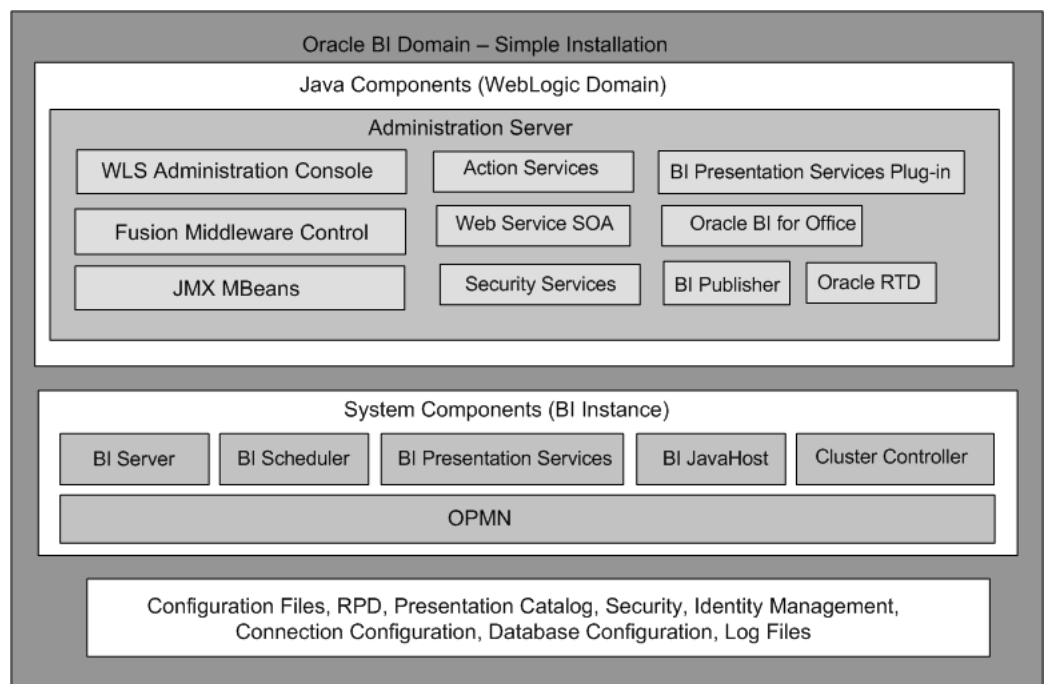
1.3.1.2 A Single Host and Simple Install

The architecture for the Simple Install type is similar to that of the Enterprise Install type, with the differences that are outlined in the following list:

- The Node Manager is not installed.
- All components that are part of the Managed Server in an Enterprise Install type are instead part of the Administration Server. The Managed Server is not installed.

Figure 1–2 illustrates the Oracle Business Intelligence system architecture on a single host for a Simple Install type.

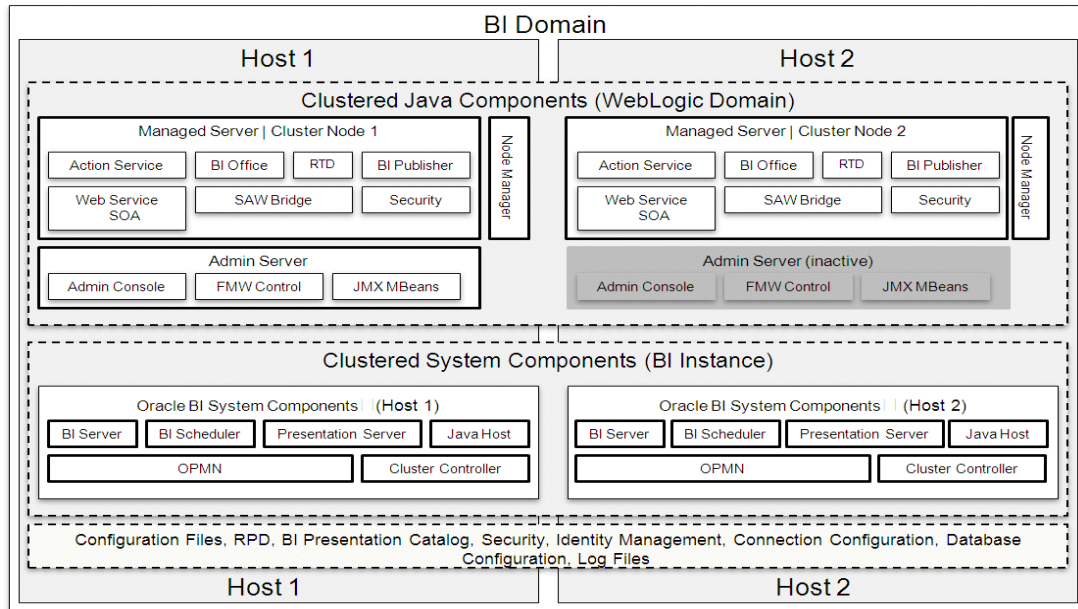
Figure 1–2 System Logical Architecture for Simple Install (Single Host)



1.3.2 Oracle Business Intelligence on Multiple Hosts

If you perform an Enterprise Installation, then Oracle Business Intelligence can be installed and configured on multiple hosts. [Figure 1–3](#) illustrates the system architecture on two hosts.

Figure 1–3 Oracle Business Intelligence System Logical Architecture (Two Hosts)



In [Figure 1–3](#), the Java components (WebLogic server 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).

1.3.3 About Java Components and System Components for Oracle Business Intelligence

The Oracle Business Intelligence Java and system components are described in the following sections:

- [Section 1.3.3.1, "About Java Components"](#)
- [Section 1.3.3.2, "About System Components"](#)

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

1.3.3.1 About Java Components

Java components are deployed as one or more Java EE applications and are described in the following list:

- **Administrative Components** — Oracle WebLogic Server Administration Console, Fusion Middleware Control, 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 for Microsoft 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. These services make it easy to invoke Oracle Business Intelligence functionality from Business Process Execution Language (BPEL) processes.
- **Oracle BI Presentation Services Plug-in** — This component is a JEE application that routes HTTP and SOAP requests to Oracle BI Presentation Services.

1.3.3.2 About System Components

System components are deployed as non-JEE components, such as processes and services written in C++ and J2SE, and are described in the following list:

- **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 an .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. It also enables Oracle BI Server query access to Hyperion Financial Management and Oracle OLAP data sources.
- **Oracle BI Cluster Controller** — This component distributes requests to the BI Server, ensuring requests are evenly load-balanced across all BI Server process instances in the BI domain.

1.3.4 What Is the Administration Server?

The Administration Server contains the administrative components that enable administration of a single or multinode (that is, distributed) BI domain, as described in the following list:

- For Enterprise and Simple Install types, the Administration Server contains the following components:

- **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 administrative user interface that provides advanced management for WebLogic, JEE components, and security.
- For Simple Install types, the Administration Server also contains the components that comprise the Managed Server in an Enterprise Install type, such as Action Services and Oracle BI Publisher.

1.4 What System Administration Tools Manage Oracle Business Intelligence?

The following sections describe system administration tools that are available to help you to manage Oracle Business Intelligence. [Table 1-2](#) outlines the tools and their purpose.

Table 1-2 System Administration Tools for Oracle Business Intelligence

Section	Tool	Purpose
Section 1.4.1	Fusion Middleware Control	Monitor, manage, and configure system components for Oracle Business Intelligence.
Section 1.4.2	Oracle WebLogic Server Administration Console	Monitor and manage JEE Java components for Oracle Business Intelligence.
Section 1.4.3	Oracle Process Manager and Notification Server (OPMN)	Manage system components for Oracle Business Intelligence (for advanced users).
Section 1.4.4	Oracle Weblogic Scripting Tool (WLST)	Programmatically administer Oracle Business Intelligence.
Section 1.4.5	Oracle BI Administration Tool	Manage the metadata repository for Oracle Business Intelligence.
Section 1.4.6	Job Manager	Manage the Oracle BI Scheduler

1.4.1 Fusion Middleware Control

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."](#)

1.4.2 Oracle WebLogic Server Administration Console

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 the health and performance of JEE servers
- Configuring WebLogic server domains
- Stopping and starting JEE servers
- Viewing JEE server logs
- Managing user populations in the LDAP Server of the Oracle WebLogic Server

For more information, see Oracle Technology Network at the following location:

<http://www.oracle.com/technology/index.html>.

1.4.3 Oracle Process Manager and Notification Server (OPMN) Tool

OPMN is a process management tool that manages the Oracle Business Intelligence system components. OPMN supports both local and distributed process management, automatic process recycling, and the communication of process state (up, down, starting, and 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 restart 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 System Components."](#)

- Automatic restart of processes when they become unresponsive or terminate unexpectedly.
- An integrated way to manage Oracle Fusion Middleware components.

1.4.4 Oracle Weblogic Scripting Tool (WLST)

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 22.2, "Accessing Oracle BI Systems Management API Methods"](#)
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

1.4.5 Oracle BI Administration Tool

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*.

1.4.6 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. For information, see *Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition*.

1.5 Working with the Sample Application

As part of the Simple Installation type, Oracle Business Intelligence is configured with a simplified "Sample Application." This application is often referred to as "SampleApp Lite." You can download and configure a more robust sample application. Instructions for this configuration are available at the following location:

<http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/overview/index.html>

See "About the SampleApp.rpd Demonstration Repository" in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about the sample repository that is provided with Oracle Business Intelligence.

1.6 Oracle BI Publisher Integration

This guide assumes that Oracle BI EE and BI Publisher have been installed and configured to run as fully integrated components at your organization. If this is not the case, then some mentions of BI Publisher in this guide might not be applicable to you. For information about running BI Publisher, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Publisher*.

1.7 Topics of Interest in Other Guides

Some topics that might be of interest to system administrators are covered in other guides. [Table 1-3](#) lists these topics and indicates where to go for more information.

Table 1–3 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"
Configuration tasks for Oracle BI Scheduler	<i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i>
Configuring data sources	<i>Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition</i>
Security, including configuring SSO and SSL	<i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>
Installing and upgrading	<i>Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence</i> <i>Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence</i>
Configuring comments and status overrides in Oracle Scorecard and Strategy Management	<i>Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition</i>
Converting Oracle Business Intelligence proprietary metadata to an XML file and importing the metadata into your Oracle or IBM DB2 database	<i>Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition</i>
Propagating UI hints (labels and tooltips) from an ADF data source to display in Oracle BI Answers	<i>Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition</i>

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

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*.

Note: You might encounter display problems when using Internet Explorer 8 with Fusion Middleware Control. For example, scroll bars might be missing on the Log Messages tab of the Diagnostics page, even when the bars are required to see all the text.

To work around this issue, ensure that Compatibility View mode is turned off for the browser. To do so:

1. From the Tools menu, select **Internet Options**. On the Advanced tab in the Browsing section, ensure that **Automatically recover from page layout errors with Compatibility View** is not checked.
 2. From the Tools menu, select **Compatibility View Settings**. Ensure that **Display intranet sites in Compatibility View** and **Display all websites in Compatibility View** are not checked.
-
-

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."](#)

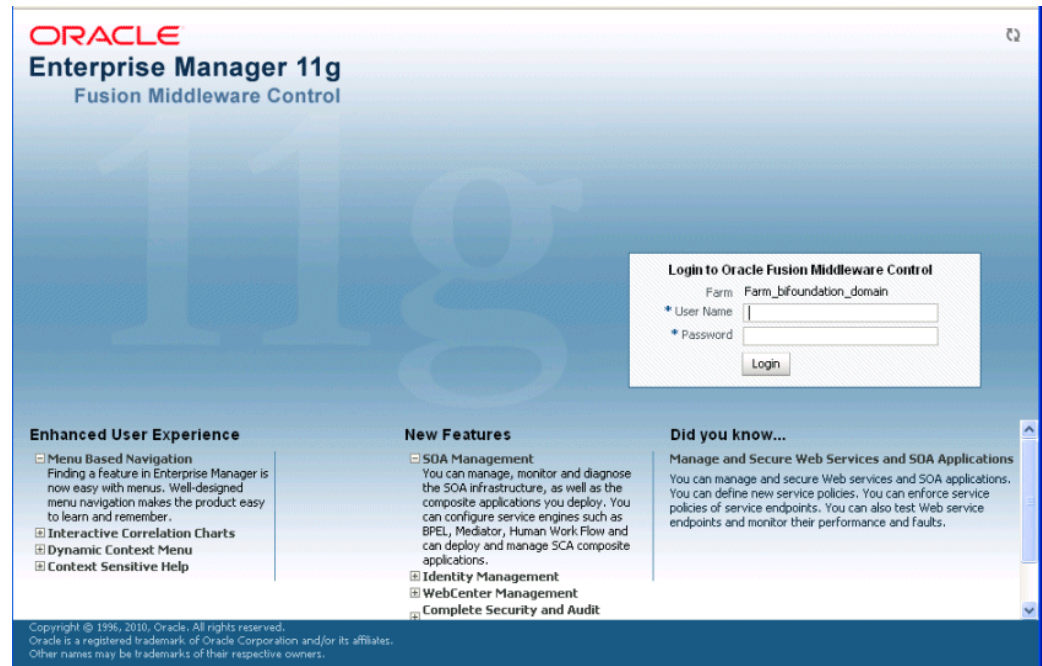
To log in to Fusion Middleware Control:

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

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

The Fusion Middleware Control login page is displayed, as shown in [Figure 2-1](#).

Figure 2-1 Sign-In Page for Fusion Middle ware Control

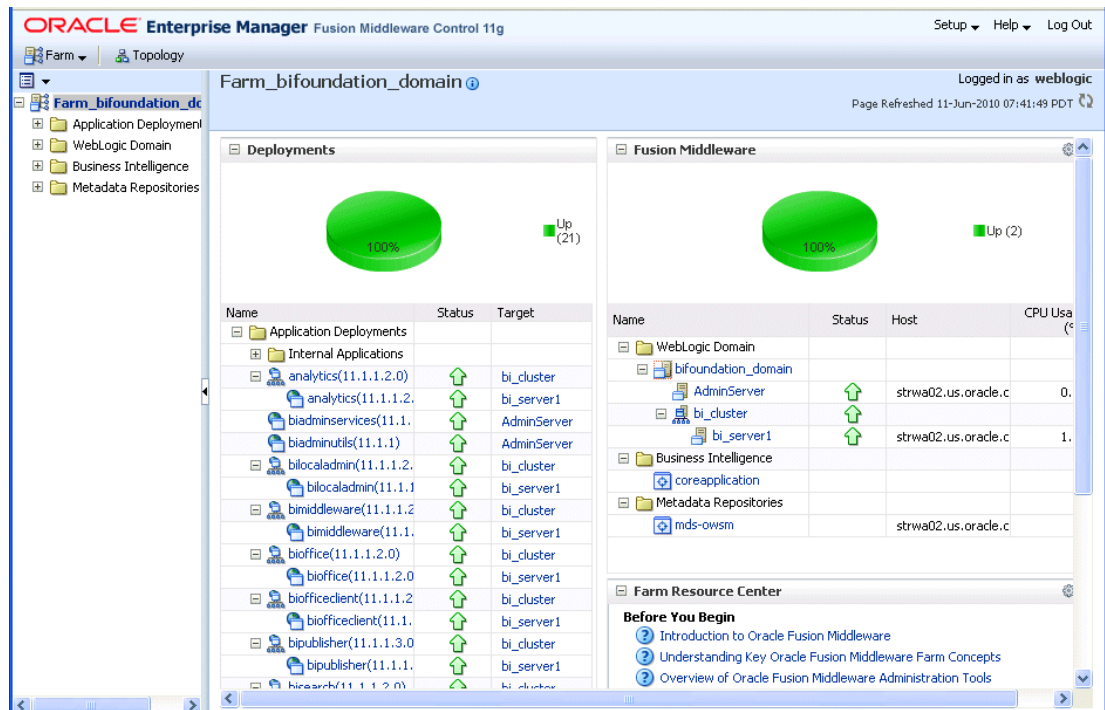


2. Enter the system administrator user name and password and click **Login**.

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 [Figure 2-2](#).

Figure 2–2 Main Page in Fusion Middleware Control

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:

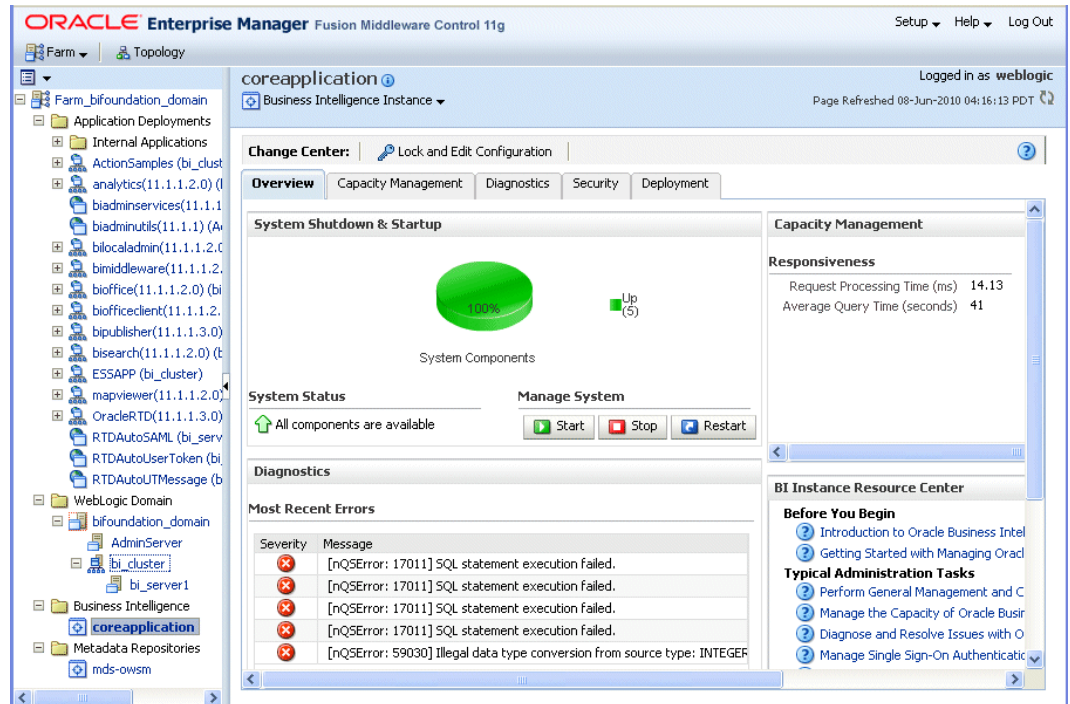
1. Log in to Fusion Middleware Control.

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 [Figure 2–3](#).

Figure 2–3 Overview Page in Fusion Middleware Control



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.

- 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 (Administration 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, as shown in [Figure 2-4](#).

Figure 2-4 Home Page for `bifoundation_domain`

The screenshot shows the Oracle Enterprise Manager 11g Fusion Middleware Control interface. The left pane shows a tree view with the `bifoundation_domain` node selected. The main pane displays the `bifoundation_domain` page, which includes a summary section with general information and a table of servers.

General Information:

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

Servers Table:

Name	Status	Host	Cluster	Listen Port	Active Sessions	Request Processing Time (ms)	Bear Accesses (per minute)
AdminServer	Up	ukp79...		7001	2	32	0.00
bi_server1	Up	ukp79...	bi_cluster	9704	0	487	0.00
ukp79575-Se...	Up	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 Administration Server and Managed Servers (`bi_server<n>`).
 - Start and stop selected Java components (Administration Server or `bi_server<n>`) using the WebLogic Domain menu Control option.
For information, see [Section 4.8.1, "Confirming If the Managed Server Is Running and Starting It."](#)
 - Manage or configure the WebLogic server 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 for which install type and components to configure, 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 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 server 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 1.3.4, "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 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 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 Sockets Layer (SSL), then you must add the letter 's' after http as follows:

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

The preceding URL example uses SSL.

The Oracle WebLogic Server Administration Console login page is displayed.

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

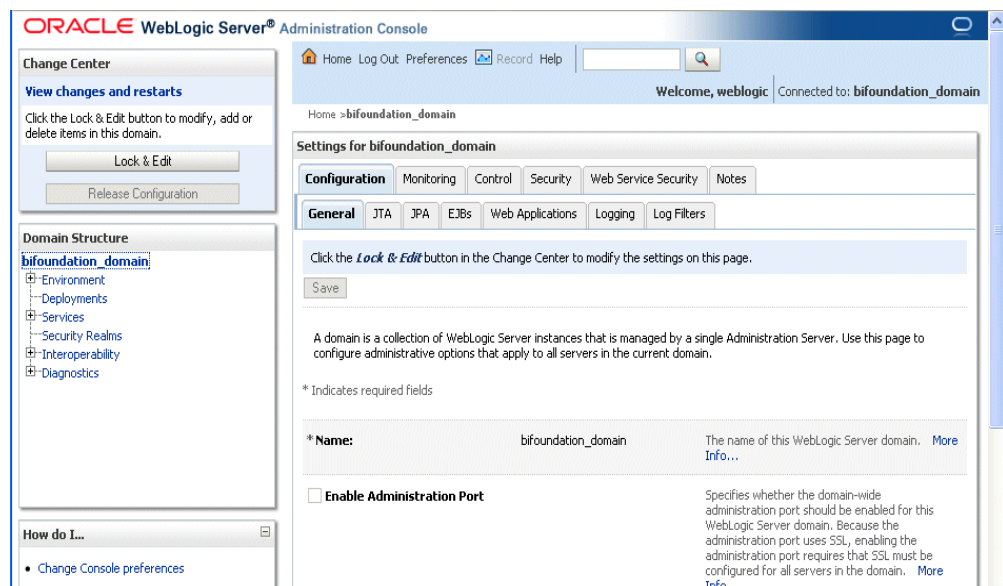
These groups provide various levels of access to system administration functions in the Oracle WebLogic Server Administration Console.

Using the security system, you can add to or delete users from 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 [Figure 2–5](#).

Figure 2–5 `bifoundation_domain` Page in Administration Console



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 1.3.4, "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.

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 includes the following topics:

- [Section 3.1, "Configuring Oracle Business Intelligence"](#)
- [Section 3.2, "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings"](#)
- [Section 3.3, "Using the Oracle BI Systems Management API to Update Configuration Settings"](#)
- [Section 3.4, "Using a Text Editor to Update Configuration Settings"](#)
- [Section 3.5, "Using the Fusion Middleware Control MBean Browser to Update Configuration Settings"](#)
- [Section 3.6, "Where Are Configuration Files Located?"](#)

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	<ul style="list-style-type: none"> ■ 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." 	<p>Automatically, through Oracle BI Systems Management API methods.</p> <p>For information, see Chapter 22, "Introducing the Oracle BI Systems Management API."</p>
Programmatically change configuration settings	<ul style="list-style-type: none"> ■ WLST (WebLogic Scripting Tool) ■ Java Program <p>For information, see Section 3.3, "Using the Oracle BI Systems Management API to Update Configuration Settings."</p>	Using scripts or programs to programmatically access Oracle BI Systems Management API methods.
Manually change configuration settings	<ul style="list-style-type: none"> ■ Text editor <p>For information, see Section 3.4, "Using a Text Editor to Update Configuration Settings."</p> <p>Caution: Settings managed by Fusion Middleware Control are commented as such in configuration files. You must not be manually update these settings.</p> <p>You must make changes correctly in configuration files in all nodes in a cluster to avoid damage to the system.</p>	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.

Note: 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 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 22.3.3, "Updating Configuration Settings Using the Oracle BI Systems Management API."](#)

3.4 Using a Text Editor to Update 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 you must add a setting 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 D, "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 its underlying node.

4. Create or update a value in 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 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\instancen\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 8.3.2, "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	<p>ORACLE_ INSTANCE\config\OracleBIServerComponent\coreapplication_ obisn</p> <p>For example: \instances\instance1\config\OracleBIServerComponent\corea pplication_obis1</p> <p>Note: Although DBFeatures.ini is also located in this directory, do not edit this file directly. See <i>Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition</i> for information about how to edit features for a database.</p>
Oracle BI Presentation Services	instanceconfig.xml	<p>ORACLE_ INSTANCE\config\OracleBIPresentationServicesComponent\core reapplication_obipsn</p> <p>For example: \instances\instance1\config\OracleBIPresentationServicesCom ponent\coreapplication_obips1</p> <p>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.</p>
Cluster Controller	ClusterConfig.xml ccslogging.xml	<p>ORACLE_ INSTANCE\config\OracleBIAApplication\coreapplication</p> <p>For example: \instances\instance1\config\OracleBIAApplication\coreapplicati on</p>
Oracle BI Scheduler	instanceconfig.xml ccslogging.xml (for Cluster Controller)	<p>ORACLE_ INSTANCE\config\OracleBISchedulerComponent\coreapplicati on_obischn</p> <p>For example: \instances\instance1\config\OracleBISchedulerComponent\core reapplication_obisch1</p>
JavaHost	config.xml logging_config.xml	<p>ORACLE_ INSTANCE\config\OracleBIJavaHostComponent\coreapplicati on_obijhn</p> <p>For example: \instances\instance1\config\OracleBIJavaHostComponent\core eapplication_obijh1</p>
Oracle BI Presentation Services Plug-in	bridgeconfig.properties	<p>MW_HOME\user_projects\domains\domain_ name\config\fmwconfig\biinstances\coreapplication</p> <p>For example: mw_home\user_projects\domains\bifoundation_ domain\config\fmwconfig\biinstances\coreapplication</p>

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 includes the following chapter:

- [Chapter 4, "Starting and Stopping Oracle Business Intelligence"](#)

Starting and Stopping Oracle Business Intelligence

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

- Section 4.1, "About Starting and Stopping Oracle Business Intelligence"
- Section 4.2, "Starting the Oracle Business Intelligence System"
- Section 4.3, "Using Fusion Middleware Control to Start and Stop Oracle Business Intelligence System Components and Java Components"
- Section 4.4, "Using Oracle WebLogic Server Administration Console to Start and Stop Java Components"
- Section 4.5, "Alternative Methods for Starting and Stopping System Components"
- Section 4.6, "Confirming Whether the Administration Server Is Running"
- Section 4.7, "Starting and Stopping the Administration Server and All Oracle Business Intelligence Java Components on Windows and UNIX"
- Section 4.8, "Starting and Stopping Managed Servers on Windows and UNIX"
- Section 4.9, "Troubleshooting System Startup"

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.

Note: This chapter provides details on starting and stopping the Oracle Business Intelligence system when an Enterprise Installation has been performed. See [Section 1.3.1, "Oracle Business Intelligence on a Single Host"](#) for information on the components that are installed with an Enterprise Installation and with a Simple Installation.

4.2 Starting the Oracle Business Intelligence System

To start the Oracle Business Intelligence system, you start the Administration Server, the Managed Servers, and the system components. The system does not include a Managed Server if the Simple Install type was selected during installation.

If the computer that hosts the Administration Server is not running or has been rebooted, then you must ensure that the computer is running and 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.

The procedure for starting the system differs slightly depending on the platform, as described in the following sections.

4.2.1 Starting Oracle Business Intelligence on Windows Platforms

To start the Oracle Business Intelligence system on Windows platforms:

1. From the Start menu, select **Programs, Oracle Business Intelligence, and Start BI Services**.

Note the following points about this command:

- For a Simple Install type, this command starts the Administration Server and Oracle Business Intelligence system and OPMN components.
- For an Enterprise or Software Only Install type, the command starts the Administration Server, system and OPMN components, Managed Server, and Node Manager. (Node Manager normally starts automatically on Windows.)
- The command does not support horizontally or vertically scaled-out Oracle WebLogic Servers, but does work on the first computer in the cluster.
- The command prompts for your credentials at first login, then stores the credentials in the boot.properties file of Oracle WebLogic Server for future use.

When you select the **Start BI Services** option, you are not prompted for administrator credentials if boot.properties files exist for both the Administration Server and for the Oracle BI Server. If either file does not exist, then you are prompted for administrator credentials and the boot.properties file is automatically created using those credentials. For information on this file see "Boot Identity Files" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

If the file is automatically created but the Administration Server does not start, then the file is deleted on the assumption that the credentials are invalid. This deletion avoids including readable text for passwords in the boot.properties file before Oracle WebLogic Server encrypts that text. However, the Administration Server can fail to start for reasons other than invalid credentials.

If the boot.properties file exists and was created by a user but the server does not start, then the file is not deleted.

4.2.2 Starting Oracle Business Intelligence on UNIX Platforms

To start the Oracle Business Intelligence system on UNIX platforms:

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. If an Enterprise or Software Only Install type was performed, then start Oracle WebLogic Node Manager manually as follows:

- a. Navigate to:

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

If a Software-Only Install type was selected during installation, then a directory other than /wlserver_10.3 might be used.

- b. 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 Node Manager Administrator's Guide for Oracle WebLogic Server*.

3. If an Enterprise or Software Only Install type was performed, then use the Oracle WebLogic Server Administration Console to check whether the Managed Server is running, and start it if necessary.

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

For information, see [Section 4.8.1, "Confirming If the Managed Server Is Running and Starting It."](#)

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 opmnctl command line

For information, see [Section 4.5.1, "Using the OPMN Command Line to Start, Stop, Restart, and View the Status of System Components."](#)

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

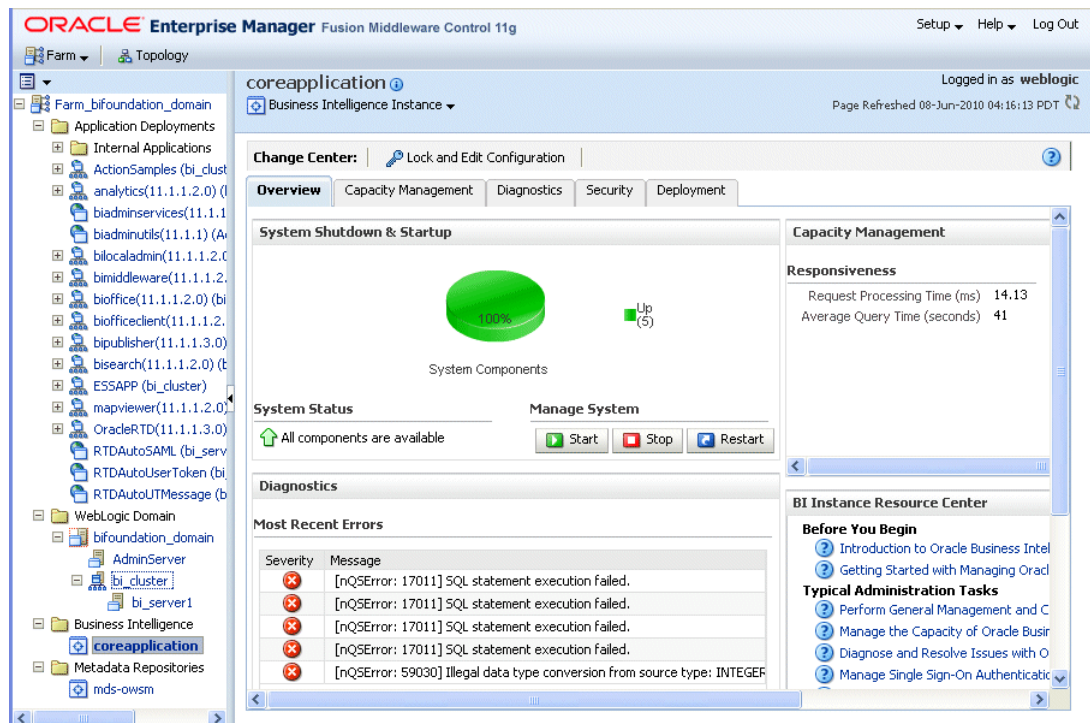
If the Oracle Business Intelligence system has been started, then you can start, stop, and restart the Oracle Business Intelligence system, including selected system components and Java components using Fusion Middleware Control.

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"](#) and shown in [Figure 4–1](#).

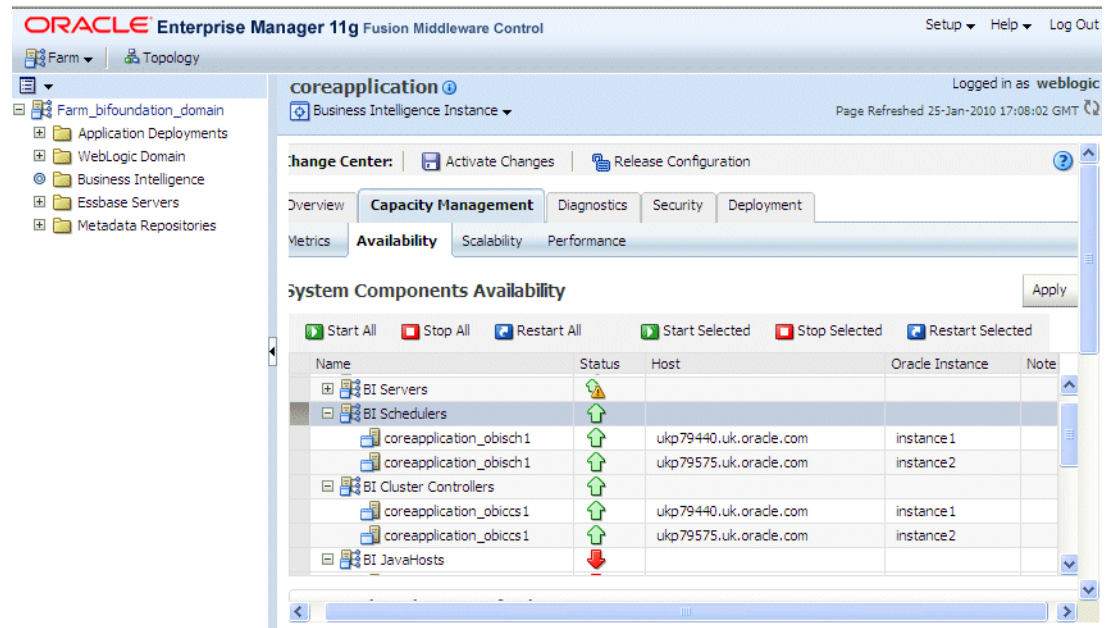
Figure 4–1 Overview Page in Fusion Middleware Control



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

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, as shown in [Figure 4–2](#).

Figure 4–2 Starting and Stopping on the Availability Tab

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 System Components"](#)

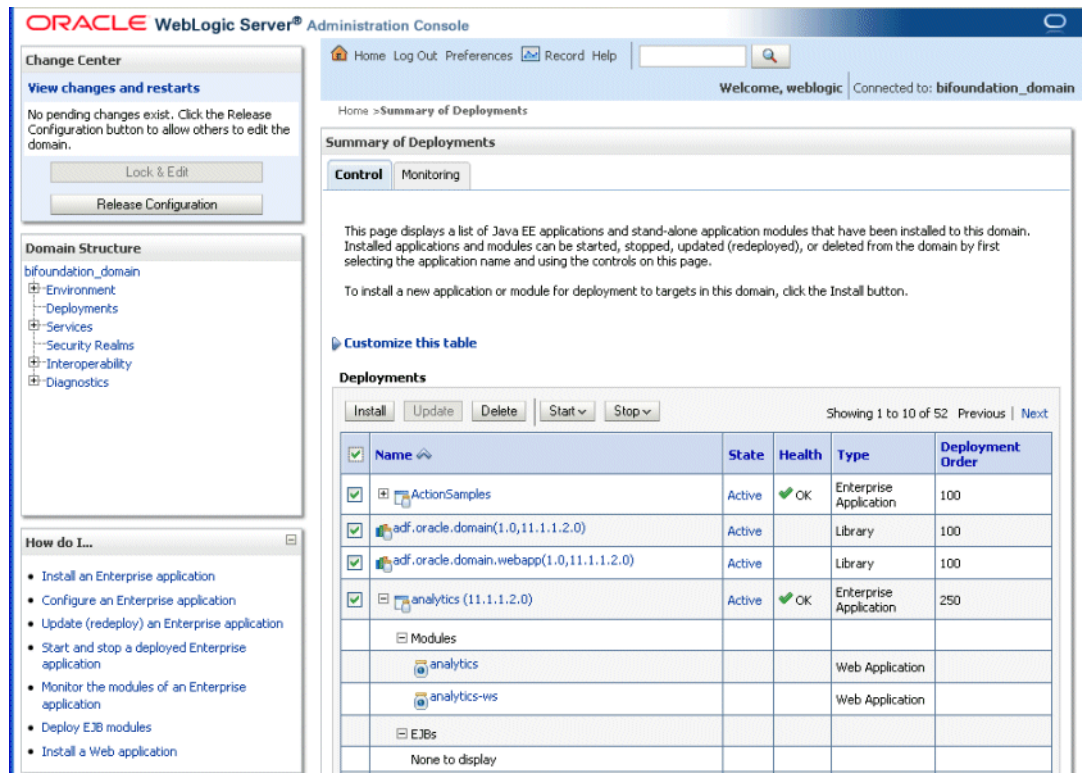
4.4 Using Oracle WebLogic Server Administration Console to Start and Stop Java Components

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, as shown in [Figure 4–3](#).

Figure 4–3 Starting and Stopping in Administration Console

4.5 Alternative Methods for Starting and Stopping 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 System Components"
- Section 4.5.2, "Using a Windows Service to Start and Stop System Components"
- Section 4.5.3, "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 System Components

You can control Oracle Business Intelligence system components across a network using `opmnctl` commands. For information about OPMN, see [Section 1.4.3, "Oracle Process Manager and Notification Server \(OPMN\) Tool."](#)

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

1. For the Oracle instance to administer, 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 [Figure 4-4](#).

Figure 4-4 Status of Components in OPMN

```
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_obijhl | 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 4.8.1, "Confirming If the Managed Server Is Running and Starting It"](#) and [Section 4.8, "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 Oracle BI Presentation Services:
`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 Oracle BI Presentation Services 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.

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 start 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.

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.

4.5.2 Using a Windows Service to Start and Stop 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 4.8.1, "Confirming If the Managed Server Is Running and Starting It"](#) and [Section 4.8, "Starting and Stopping Managed Servers on Windows and UNIX."](#)

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 5 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 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 22.3.2, "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):
- 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.cmd -start
```

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

```
./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). For more information, see "Setting Up a WebLogic Server Instance as a Windows Service" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

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**.
- 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 Starting and Stopping Managed Servers on Windows and UNIX

You can start and stop Managed Servers on Windows and UNIX. The system does not include a Managed Server if the Simple Install type was selected during installation.

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. For more information, see "Setting Up a WebLogic Server Instance as a Windows Service" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

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.

4.8.1 Confirming If the Managed Server Is Running and Starting It

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. The system does not include a Managed Server if the Simple Install type was selected during installation.

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 4.8, "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 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 System Components."](#)

4.9 Troubleshooting System Startup

This section contains solutions that are related to system startup:

- [Section 4.9.1, "Administration Server Fails to Start When the Database Is Not Running"](#)
- [Section 4.9.2, "Managed Server Is Down"](#)
- [Section 4.9.3, "Oracle BI Server Fails to Start"](#)
- [Section 4.9.4, "Oracle BI Presentation Services Fails to Start"](#)
- [Section 4.9.5, "Cannot Log In"](#)

4.9.1 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.

4.9.2 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 4.8.1, "Confirming If the Managed Server Is Running and Starting It."](#)

4.9.3 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.

4.9.4 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\coreapplicat
ion_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 System Components."](#)
- **Problem:** Authentication failed for BISystemUser: invalid user name and password.

Cause: The Managed Server is not running.

Resolution: See [Section 4.9.2, "Managed Server Is Down."](#)

4.9.5 Cannot Log In

- **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 includes the following chapters:

- [Chapter 5, "Scaling Your Deployment"](#)
- [Chapter 6, "Deploying Oracle Business Intelligence for High Availability"](#)
- [Chapter 7, "Managing Performance Tuning and Query Caching"](#)

Scaling Your Deployment

This chapter describes how to manage the capacity of your deployment. By default for an Enterprise Install or Software Only Install type, the Oracle Business Intelligence components are installed in a cluster configuration and are scalable. (The Simple Install type does not support scaling.) 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 includes 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 5.4, "Setting Up Shared Files and Directories"](#)
- [Section 5.5, "Using Fusion Middleware Control to Scale System Components"](#)
- [Section 5.6, "Validating That Your System Has Been Scaled Correctly"](#)

Note: Because the Simple Install type deploys Oracle Business Intelligence components in the Administration Server, you cannot scale out a deployment that is based on the Simple Install type. Use an Enterprise Install or Software Only Install type for production deployments.

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 1.3.3, "About Java Components and System Components for Oracle Business Intelligence"](#) 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 a 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:

- Ensure that you 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 run 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 Installation 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 select the Enterprise Install option	"Enterprise Install to Scale Out Existing Installations" in <i>Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence</i>
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	<i>Oracle Fusion Middleware Using Web Server Plug-Ins with Oracle WebLogic Server</i> "Load Balancing in a Cluster" in <i>Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server</i>

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 10.2, "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 10.2, "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 19.3.3.1, "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

The screenshot displays the 'Scalability' tab within the 'Capacity Management' section of Fusion Middleware Control. The page title is 'coreapplication' and the user is logged in as 'weblogic'. The page was refreshed on Jul 13, 2010 at 4:09:31 PM PDT. The 'Change Center' shows 'Activate Changes' and 'Release Configuration' options. The 'Capacity Management' section is active, with sub-tabs for 'Metrics', 'Availability', 'Scalability', and 'Performance'. The 'Scalability' sub-tab is selected, and an 'Apply' button is visible. The main content area is titled 'System Components' and includes the instruction: 'Edit the table below to increase/decrease/modify the system components for each instance.' The table has the following data:

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

Below the table, there is a section for 'JEE Components' with the text: 'By default JEE components are symmetrically distributed across every BI Instance in the domain.' Two links are provided: 'Go to the Oracle WebLogic Server Administration Console to configure and manage JEE components.' and '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 about using methods in the Oracle BI Systems Management API to scale out components, see [Chapter 22, "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.

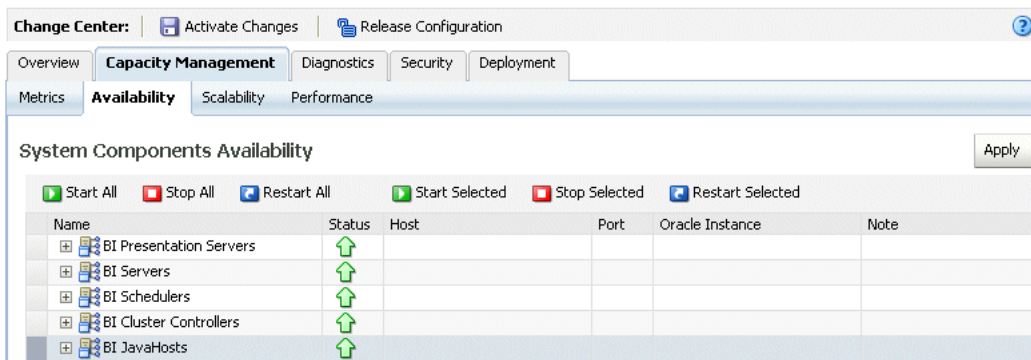
On this page, you can:

- 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



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

Summary of Servers

Configuration Control

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.

[Customize this table](#)

Servers (Filtered - More Columns Exist)

New Clone Delete Showing 1 to 2 of 2 Previous | Next

<input type="checkbox"/>	Name	Cluster	Machine	State	Health	Listen Port
<input type="checkbox"/>	AdminServer(admin)		adc2190722	RUNNING	✔ OK	7001
<input type="checkbox"/>	bi_server1	bi_cluster	adc2190722	RUNNING	✔ OK	9704

New Clone Delete Showing 1 to 2 of 2 Previous | Next

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 High Availability for Oracle Business Intelligence and EPM" 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

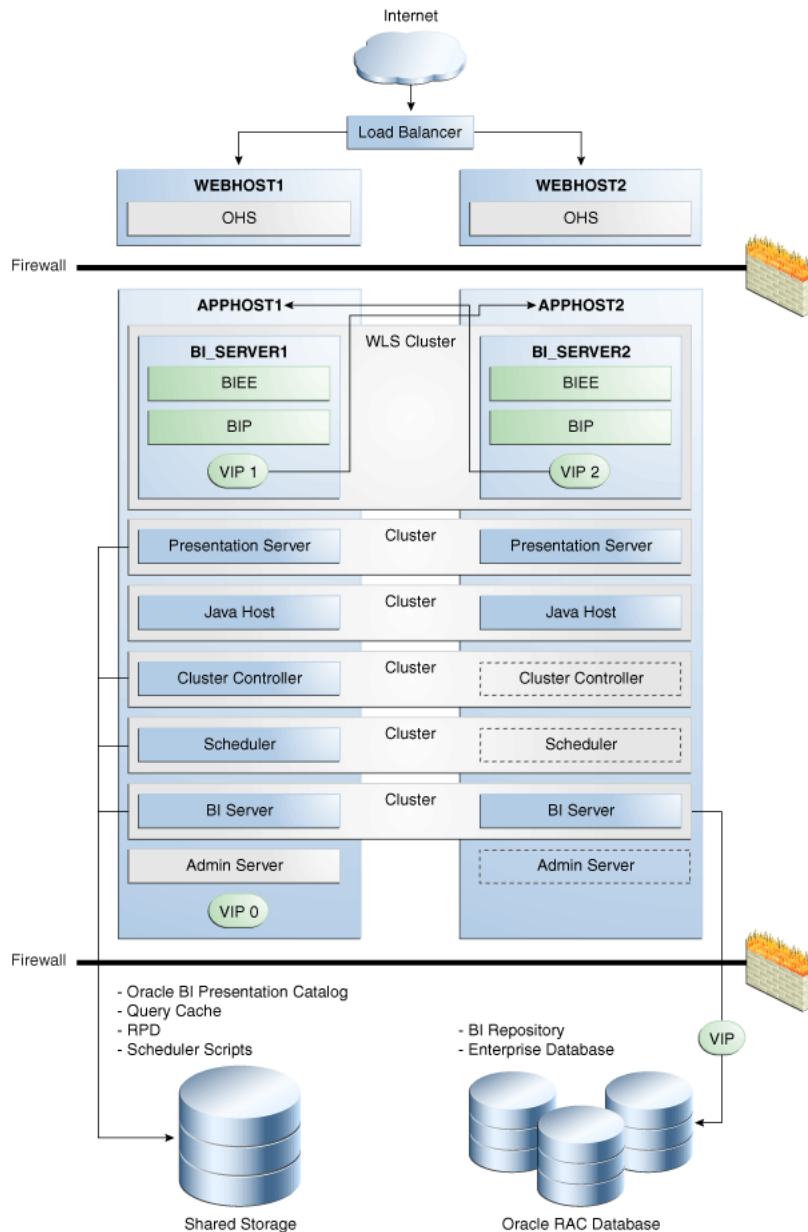
This chapter includes the following topics:

- [Section 6.1, "About Oracle Business Intelligence Components in a Clustered Environment"](#)
- [Section 6.2, "Configuring Oracle Business Intelligence Components for High Availability"](#)
- [Section 6.3, "Optional Configuration for Oracle Business Intelligence High Availability"](#)
- [Section 6.4, "Using the Cluster Manager"](#)
- [Section 6.5, "Troubleshooting an Oracle Business Intelligence Clustered Environment"](#)

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 1.3.3, "About Java Components and System Components for Oracle Business Intelligence" 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 High Availability for Oracle Business Intelligence and EPM" 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.

6.1.3 Achieving High Availability Using an Active-Passive Model

As an alternative to setting up the active-active configuration described in the previous sections, you can set up Oracle Business Intelligence in an active-passive configuration using Oracle Fusion Middleware Cold Failover Cluster (Cold Failover Cluster). In a Cold Failover Cluster configuration, two or more application server instances are configured to serve the same application workload, but only one is active at any particular time.

A two-node Cold Failover Cluster can be used to achieve active-passive availability for Oracle Business Intelligence. In a Cold Failover Cluster, one node is active while the other is passive, on standby. In the event that the active node fails, the standby node is activated, and Oracle Business Intelligence continues servicing clients from that node. All Oracle Business Intelligence components are failed over to the new active node. No Oracle Business Intelligence components run on the failed node after the failover.

See "Active-Passive Topologies for Oracle Fusion Middleware High Availability" in *Oracle Fusion Middleware High Availability Guide* for detailed information.

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 that 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

The screenshot shows the Fusion Middleware Control interface for a Business Intelligence Instance. The main navigation includes Overview, Capacity Management (selected), Diagnostics, Security, and Deployment. Under Capacity Management, the Availability tab is active, showing a table of System Components Availability. Below this, there is a section for Potential Single Points of Failure and a section for Primary/Secondary Configuration.

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

Risk of failure	Name	Type	Recommended 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

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 about using methods in the Oracle BI Systems Management API to manage availability, see [Chapter 22, "Introducing the Oracle BI Systems Management API."](#)

6.3 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 6.3.1, "Setting Optional Cluster Controller Parameters"](#)
- [Section 6.3.2, "Setting Optional Presentation Services Parameters"](#)
- [Section 6.3.3, "Setting Optional Oracle BI Presentation Services Plug-in Parameters"](#)

6.3.1 Setting Optional Cluster Controller Parameters

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

A copy of the ClusterConfig.xml 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 ClusterConfig.xml file:

1. Open the ClusterConfig.xml file for editing. You can find the file at:
`ORACLE_INSTANCE/config/OracleBIApplication/coreapplication`
2. [Table 6–2](#) describes default values for the cluster communication parameters under the ClusterProperties element. Optionally, modify the parameter values as required for the deployment.

Table 6–2 ClusterConfig.xml Parameters for Cluster Communication

Parameter	Description	Default Value
ServerPollSeconds	The frequency of heartbeat messages between the Cluster Controller and the Oracle BI Server and Oracle BI Scheduler nodes in the cluster.	5 seconds
ControllerPollSeconds	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 ClusterConfig.xml file. Note that any additional elements that are not shown in this example are centrally managed and cannot be set manually.

Example 6-1 Sample Parameters for Clustering in ClusterConfig.xml

```
<ClusterProperties>
  <ServerPollSeconds>5</ServerPollSeconds>
  <ControllerPollSeconds>5</ControllerPollSeconds>
</ClusterProperties>
```

6.3.2 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_obipsn
2. Under the ServerInstance tag, the JavaHostProxy element has optional subelements. [Table 6-3](#) describes these optional subelements.

Table 6-3 Optional Subelements for the JavaHostProxy Element

Subelement	Attribute	Description
LoadBalancer/Ping	keepAliveMaxFailures	Specifies the number of ping failures required before the host is declared nonfunctioning. 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.

6.3.3 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.

6.4 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**, then **Clusters**.
3. Select **Refresh**, then **Every**, and select 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**, then **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**, then **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**.

6.4.1 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.

6.4.1.1 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**, then **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.
Role	The role of the object in the cluster: <ul style="list-style-type: none"> ■ 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.

Table 6–4 (Cont.) Status Columns

Column	Description
Status	<p>The status of the object in the cluster:</p> <ul style="list-style-type: none"> ■ Online. The Cluster Controller or Oracle BI Server is online. Online Cluster Controllers can accept session requests and assign them to available servers within the cluster. Online Oracle BI Servers can be assigned sessions by the Cluster Controller. ■ Quiesce. This status is applicable to clustered servers only. When a server is quiesced, 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. Offline Cluster Controllers cannot accept session requests or assign sessions to available servers within the cluster. Offline Oracle BI Servers do not communicate 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).
Type	<p>When Clusters is selected in the Explorer pane, this field is available. There are three types:</p> <ul style="list-style-type: none"> ■ Controller. The object is a Cluster Controller. ■ Server. The object is an Oracle BI Server. ■ Scheduler. The object is a Scheduler Server.

6.4.1.2 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 is smaller than Full size.

Table 6–5 (Cont.) Cache View Columns

Column	Description
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:

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

6.4.1.3 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 is displayed 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 is displayed 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.

Table 6–7 (Cont.) Request Window Columns (Bottom Window)

Column	Description
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	<p>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.</p> <ul style="list-style-type: none"> ▪ 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 an option.

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 select **View**, then **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.

6.4.1.4 Server Information

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

6.5 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 an incorrect configuration. Although 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.

6.5.1 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 later.

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
```

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 must be paid only once for a query, not every time the query is run.

See also the following Oracle Fusion Middleware resources on performance tuning for your system:

Oracle Fusion Middleware Performance and Tuning Guide

Oracle Fusion Middleware Performance and Tuning for Oracle WebLogic Server

This chapter includes the following topics:

- [Section 7.1, "Monitoring Service Levels"](#)
- [Section 7.2, "About Query Performance Tuning"](#)
- [Section 7.3, "Setting Performance Parameters in Fusion Middleware Control"](#)
- [Section 7.4, "About the Oracle BI Server Query Cache"](#)
- [Section 7.5, "Configuring Query Caching"](#)
- [Section 7.6, "Monitoring and Managing the Cache"](#)
- [Section 7.7, "Strategies for Using the Cache"](#)
- [Section 7.8, "Cache Event Processing with an Event Polling Table"](#)
- [Section 7.9, "Managing the Oracle BI Presentation Services Cache Settings"](#)
- [Section 7.10, "Improving Oracle BI Web Client Performance"](#)

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 WSLT 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. If your deployment is based on the Simple Install type, use the Monitoring tab for the Administration Server.

To view metrics for Oracle Business Intelligence in the Monitoring tab:

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 server name (for example, **oracle_bi1** or **AdminServer(admin)**).
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 quickly, 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 the database administrator (DBA) for analysis. See [Section 8.4, "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 several 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 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 about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "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 about using methods in the Oracle BI Systems Management API to change settings, see [Chapter 22, "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 18.3, "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** option
 - **Maximum Number of Rows Per Page to Include** option

The value of these options cannot exceed the value that is specified for the MaxVisibleRows element, as described in [Table 18–1](#).

5. Click **Apply**, then click **Activate Changes**.
6. Return to the Business Intelligence Overview page and click **Restart**.

For information about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "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 18.3, "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 is likely to 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 about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "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 11g 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 the beginning 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 usually 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.

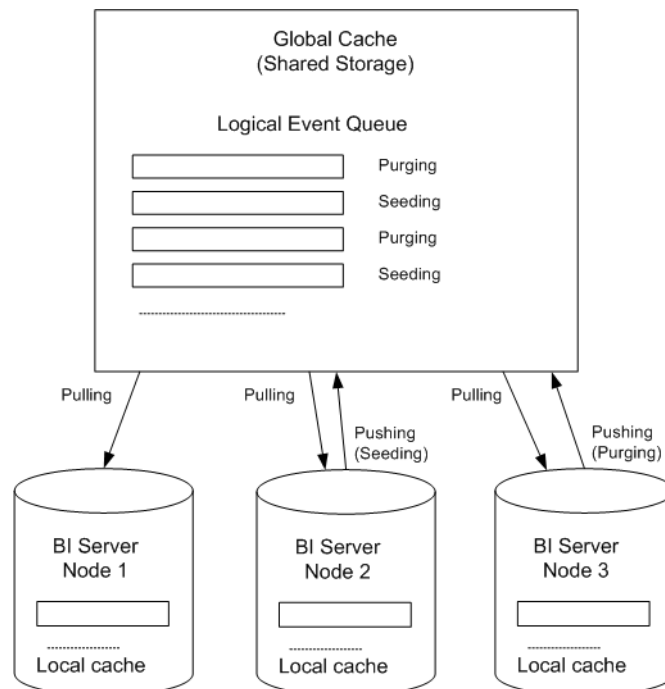
For more information about XML data sources, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

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 about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "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 about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "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 about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "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 `nquery.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 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 when any of the ODBC procedures to purge the cache are called. 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 quotation mark within the string argument of a procedure, then you must use another single quotation mark 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 quotation marks that are 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:

```
SAPurgeALLMCNCache ()
```

- **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' )
```

Table 7–2 describes the messages that are 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.

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 are 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	<p>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.</p> <p>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.</p>
Columns in the SELECT list can be composed of expressions on the columns of the cached queries	<p>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:</p> <pre>SELECT product, month, averageprice FROM sales WHERE year = 2000</pre> <p>hits cache on the query:</p> <pre>SELECT product, month, dollars, unitsales FROM sales WHERE year = 2000</pre> <p>because averageprice can be computed from dollars and unitsales (averageprice = dollars/unitsales).</p>
WHERE clause must be semantically the same or a logical subset	<p>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.</p> <p>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:</p> <ul style="list-style-type: none"> ■ 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: <pre>SELECT employeename, region FROM employee, geography WHERE region in ('EAST', 'WEST')</pre> qualifies as a hit on the following cached query: <pre>SELECT employeename, region FROM employee, geography WHERE region in ('NORTH', 'SOUTH', 'EAST', 'WEST')</pre> ■ 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: <pre>WHERE revenue < 1000</pre> qualifies as a cache hit on a comparable query with the predicate: <pre>WHERE revenue < 5000</pre> ■ 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	<p>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.</p>
Queries with special functions must be an exact match	<p>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.</p>
Set of logical tables must match	<p>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.</p>

Table 7–3 (Cont.) Factors That Determine Whether Cache Is Hit

Factor or Rule	Description
Session variable values must match, including security session variables	<p>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.</p> <p>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 Section 7.7.1.1, "Ensuring Correct Cache Results When Using Row-Level Database Security" for more information.</p>
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, <code>SELECT DISTINCT...</code>), 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	<p>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:</p> <pre>SELECT supplier, region, city, qtysold FROM suppliercity</pre> <p>The following query requests the quantity sold at the city level:</p> <pre>SELECT city, qtysold FROM suppliercity</pre> <p>The second query results in a cache hit on the first query.</p>
Limited additional aggregation	For example, if a query with the column <code>qtysold</code> is cached, then a request for <code>RANK (qtysold)</code> results in a cache miss. Additionally, a query that requests <code>qtysold</code> at the country level can get a cache hit from a query that requests <code>qtysold</code> at the country, region level.
ORDER BY clause must be comprised 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 <code>USE_ADVANCED_HIT_DETECTION</code> to <code>YES</code> in the <code>NQConfig.INI</code> 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 to populate 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**, then **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.

You can control the cache entry information and its display sequence using the Options settings (select **Edit**, then select **Options** from the Cache Manager, or select **Tools**, then **Options**, then **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 is 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 <code>MAX_CACHE_ENTRIES</code> parameter in the <code>NQSCONFIG.INI</code> 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 <code>MAX_ROWS_PER_CACHE_ENTRY</code> parameter in the <code>NQSCONFIG.INI</code> 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**, then **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**, then **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**, then **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**, then **Refresh** or press F5 to refresh the 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 (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 <code>DEFAULT CURRENT_TIMESTAMP</code> 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 <code>FRACTIONAL_SECOND_PRECISION</code> 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 <code>FRACTIONAL_SECOND_PRECISION</code> . For example, for Oracle Database or Teradata, you might want to change <code>FRACTIONAL_SECOND_PRECISION</code> 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, to specify any explicit storage clauses, 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. From the Tools menu, select **Utilities**.
2. Select the option **Oracle BI Event Tables** from the list of options.
3. Click **Execute**.
4. Select the table 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 `nquery.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 `nquery.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_obipsn\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 different from the 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 about 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 about 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 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

You can improve the performance of the Oracle BI Web client by configuring your Web server to serve up all static files, as well as enabling compression for both static and dynamic resources. By enabling caching and content expiration on the Web server, Web browsers can determine how often to reload the static files from the server.

In Oracle BI EE, static files are located in `ORACLE_HOME/bifoundation/web/app`. Follow the instructions for your Web server to set up static file caching and compression for the files located in this directory.

Note: See the following documents for full information about how to configure Oracle WebLogic Server to work with Web servers like Apache HTTP Server, Microsoft Internet Information Server (Microsoft IIS), and Oracle HTTP Server:

Oracle Fusion Middleware Using Web Server 1.1 Plug-Ins with Oracle WebLogic Server

Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server

The following sections provide example configurations:

- [Section 7.10.1, "Configuring Apache HTTP Server for Static File Caching"](#)
- [Section 7.10.2, "Configuring Oracle HTTP Server for Static File Caching"](#)

7.10.1 Configuring Apache HTTP Server for Static File Caching

This example configuration assumes that you have installed the Web server plug-in that allows Apache HTTP Server to proxy requests to Oracle WebLogic Server. Make sure that the `PLUGIN_HOME/lib` directory is added to `LD_LIBRARY_PATH`, or equivalent for your operating system.

The steps in this section show an example configuration only. You can adjust your configuration as needed. See *Oracle Fusion Middleware Using Web Server 1.1 Plug-Ins with Oracle WebLogic Server* for full information.

To add configuration directives for the plug-in:

1. Locate the `httpd.conf` file for your Apache HTTP Server.
2. Open the file for editing and add directives similar to the following:

```
LoadModule weblogic_module modules/mod_wl.so

<IfModule mod_weblogic.c>
    WebLogicPort 9704
    Debug OFF
    WebLogicHost localhost
    WLLogFile /tmp/wl-proxy.log
</IfModule>

<LocationMatch "/analytics/saw\.dll.*">
    SetOutputFilter DEFLATE
    SetHandler weblogic-handler
</LocationMatch>

<LocationMatch "/analytics/.*\.jsp.*">
    SetOutputFilter DEFLATE
    SetHandler weblogic-handler
```

```
</LocationMatch>
```

Note the following:

- Modify the LoadModule directive based on where and how you installed the plug-in.
 - The IfModule directive enables the connection to Oracle WebLogic Server. See *Oracle Fusion Middleware Using Web Server 1.1 Plug-Ins with Oracle WebLogic Server* for more information about the connectivity options, including how to configure a cluster and SSL considerations.
 - The LocationMatch directives are used to route all dynamic requests to Oracle WebLogic Server. Be sure to include the SetOutputFilter DEFLATE directive, which enables GZIP compression for all dynamic requests.
3. Save and close the file.

To add configuration directives for handling static files:

1. Locate the httpd.conf file for your Apache HTTP Server.
2. Open the file for editing and add directives similar to the following:

```
Alias /analytics ORACLE_HOME/bifoundation/web/app
<Directory ORACLE_HOME/bifoundation/web/app>
# Disable cross-server ETags
FileETag none
# Enable compression for all files
SetOutputFilter DEFLATE
# Don't compress images
SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
# Enable future expiry of static files
ExpiresActive on
ExpiresDefault "access plus 1 week"
DirectoryIndex default.jsp
</Directory>

# Restrict access to WEB-INF
<Location /analytics/WEB-INF>
Order Allow,Deny
Deny from all
</Location>
```

Note the following:

- You must ensure that Apache HTTP Server has access to the static files for the Oracle BI Web client in *ORACLE_HOME/bifoundation/web/app*. Ensure that the Web server is running and has read access to this location.
- The Alias and Directory entries tell Apache HTTP Server to handle requests for static files rather than routing them to Oracle WebLogic Server. Note the following about the directives related to compression and static file expiry:

- FileETag

```
FileETag none
```

This directive tells the Web server to disable generation of ETag headers in the response. Default ETag generation for Apache HTTP Server is tied to the file system for a single server, so generating ETags is not recommended.

- Compression Related Directives

```

SetOutputFilter DEFLATE
# Don't compress images
SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary

```

These directives ensure that Apache HTTP Server compresses all files except images. Typically, images are already compressed and do not benefit from additional compression.

– Control of Expires Header

```

# Enable future expiry of static files
ExpiresActive on
ExpiresDefault "access plus 1 week"

```

This fragment tells Apache HTTP Server to enable setting the Expires header. In this example, the default expiration is set to one week after the first time the file was accessed by the client. You can increase this time period, but ensure that static files are refreshed often enough to handle any patches or updates made on the static files.

– Handling Default URLs

```

DirectoryIndex default.jsp

```

This directive provides a fallback handler when a user requests the /analytics URL without specifying any content under it. This URL is then routed to Oracle WebLogic Server for further processing.

- The final directive restricts access to the WEB-INF folder. This folder is part of the J2EE container's deployment descriptor and must not be exposed to Web clients.

3. Save and close the file.

7.10.2 Configuring Oracle HTTP Server for Static File Caching

Configuration for Oracle HTTP Server is similar to configuration for Apache HTTP Server, except that you do not need to download and install the plug-in because the `mod_wl_ohs.so` module is installed by default with Oracle HTTP Server. Some configuration is performed in the `mod_wl_ohs.so` module directly, and some configuration is performed in `httpd.conf`. See *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server* for full information.

Part IV

Resolving Issues

This part explains how to resolve issues in Oracle Business Intelligence. It includes the following chapters:

- [Chapter 8, "Diagnosing and Resolving Issues in Oracle Business Intelligence"](#)
- [Chapter 9, "Managing Usage Tracking"](#)

Diagnosing and Resolving Issues in Oracle Business Intelligence

This chapter describes how to diagnose and resolve issues and includes the following topics:

- [Section 8.1, "How Are Issues Resolved?"](#)
- [Section 8.2, "Viewing And Configuring Diagnostic Log Files"](#)
- [Section 8.3, "Understanding Diagnostic Log and Log Configuration Files"](#)
- [Section 8.4, "Managing the Query Log"](#)
- [Section 8.5, "Logging in Oracle BI Presentation Services"](#)

8.1 How Are Issues Resolved?

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 8.2, "Viewing And Configuring Diagnostic Log Files"](#)
 - [Section 8.3, "Understanding Diagnostic Log and Log Configuration Files"](#)
 - [Section 8.4, "Managing the Query Log"](#)
 - [Section 8.5, "Logging in Oracle BI Presentation Services"](#)
- Examination of system and process metrics to understand availability and performance issues. For information, see:
 - [Chapter 7, "Managing Performance Tuning and Query Caching"](#)

8.2 Viewing And Configuring Diagnostic Log Files

You can view diagnostic log files and configure settings that affect diagnostic log files and the information that they contain, as described in the following sections:

- [Section 8.2.1, "Using Fusion Middleware Control to View Log Information, Error Messages, and Alerts"](#)
- [Section 8.2.2, "Configuring Log File Rotation Policy and Specifying Log Levels"](#)

8.2.1 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.

8.2.2 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 the log files contain.

This section contains the following topics:

- [Section 8.2.2.1, "Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels"](#)
- [Section 8.2.2.2, "Manually Changing Additional Log File Settings"](#)

For information about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "Introducing the Oracle BI Systems Management API."](#)

8.2.2.1 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**.

8.2.2.2 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 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 Configuration Settings."](#)

To manually change the settings that configure the log file format:

1. Open the component log configuration file as described in [Section 8.3.2, "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.

8.3 Understanding Diagnostic Log and Log Configuration Files

This section discusses diagnostic log files and diagnostic log configuration files, and contains the following topics:

- [Section 8.3.1, "What Are Diagnostic Log Files and Where Are They Located?"](#)
- [Section 8.3.2, "What Are Diagnostic Log Configuration Files and Where Are They Located?"](#)
- [Section 8.3.3, "What Are Log File Message Categories and Levels?"](#)
- [Section 8.3.4, "What is Log File Rotation?"](#)

8.3.1 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\sawcatalogcrawlerlogsysn.log` — The catalog crawler log file, which is not searchable in the Fusion Middleware Control Log Viewer.
 - `sawlogn.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 8.5, "Logging in Oracle BI Presentation Services."](#)

- Oracle BI Server
 - `nquery<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 `nquery-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.

8.3.2 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 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 the system writes to the `sawlog.log` file.
- Oracle BI Server log configuration file:

- Log settings configure messages that the system writes to the nqserver.log file.

For more information, see [Section 8.3.5, "What Messages Are Included in the System Log?"](#)

- UserLog settings configure messages that the system writes to the nquery.log file.

For more information, see [Section 8.4, "Managing the Query Log."](#)

- Oracle BI Scheduler log configuration file:
 - Log settings configure messages that the system writes to the nqscheduler.log file.
- JavaHost Server log configuration file:
 - log_handlers elements and subelements enable configuration of the log file rotation policy and the specification of the log file name and its location.
 - loggers elements and subelements 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.

8.3.3 What Are Log File Message Categories and Levels?

Categories and levels for log file messages define the detail and level of importance with which the system writes messages 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.

Table 8–1 (Cont.) Log File Message Category Levels

Category:Level	Description
Trace:1	<p>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.</p> <p>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.</p>
Trace:16	<p>A fairly detailed trace or debug message has been written. The message is clear enough to be understood by Oracle Support Services engineers who have a deep knowledge of the product but might not know full details of the internal implementation.</p> <p>High performance impact. This level should not be enabled on a production environment, except on special situations to debug issues with the software.</p>
Trace:32	<p>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.</p> <p>Very high performance impact. This level is not expected to be enabled in a production environment and developers use it only to debug the software on a test or development environment.</p>

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 8.2.2.1, "Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels."](#)

8.3.4 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 8.2.2.1, "Using Fusion Middleware Control to Configure Log File Rotation Policy and Specify Log Levels."](#)

8.3.5 What Messages Are Included in the System Log?

The Oracle BI Server writes messages to the `nqserver.log` file, based on configuration settings. In addition to writing messages to this log file, the BI Server writes certain severe messages to the system log file for UNIX systems. The following list includes the kinds of messages that the BI Server writes to the system log file:

- When the BI Server cannot start (for example, because another server has previously started), then the system log file includes a message such as the following one:
Another server is already running on : @1%ls and port: @2%ls.
- When memory problems occur, the system log file includes a message such as the following one:
Could not enable the Low-Fragmentation Heap.
- When the hard disk on the computer is full, the system log file includes a message such as the following one:
Out of disk space.

8.4 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 Services. 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, 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, in the User tab. In the Administration Tool, select **Identity** from the Manage option on the main toolbar. In the Identity Manager dialog, double-click a user and select the User tab.

This section contains the following topics:

- [Section 8.4.1, "Configuring Query Logging"](#)
- [Section 8.4.2, "Using the Log Viewer"](#)

8.4.1 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.

8.4.1.1 Setting the Query Logging Level

You can enable query logging levels for individual users, as described in [Section 8.4.1.2, "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 is appropriate for your organization. In normal operations, logging is generally disabled (that is, the logging level is set to 0). If you decide to enable logging, then select 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	<p>Logs the SQL statement issued from the client application. Also logs the following:</p> <ul style="list-style-type: none"> ■ 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.
Level 2	<p>Logs everything logged in Level 1.</p> <p>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.</p>
Level 3	<p>Logs everything logged in Level 2.</p> <p>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.</p> <p>Do not select this level without the assistance of Oracle Support Services.</p>
Level 4	<p>Logs everything logged in Level 3.</p> <p>Additionally, logs the query execution plan. Do not select this level without the assistance of Oracle Support Services.</p>
Level 5	<p>Logs everything logged in Level 4.</p> <p>Additionally, logs intermediate row counts at various points in the execution plan. Do not select this level without the assistance of Oracle Support Services.</p>
Level 6 and 7	Not used.

8.4.1.2 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 Identity Manager dialog is displayed.
2. Double-click the name of the user for which you want to set the query logging level.
The User dialog 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**.

8.4.2 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.

8.4.2.1 Running the Log Viewer Utility

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]
```

In this syntax:

- *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.

8.4.2.2 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 database administrators (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.

8.5 Logging in Oracle BI Presentation Services

This section describes logging specifically in Presentation Services and contains the following topics:

- [Section 8.5.1, "Using the Oracle BI Presentation Services Logging Facility"](#)
- [Section 8.5.2, "Structure for the Oracle BI Presentation Services Configuration File"](#)
- [Section 8.5.3, "Examples of the Formats of Logged Messages"](#)
- [Section 8.5.4, "Oracle BI Presentation Services Message Structure"](#)

- [Section 8.5.5, "Oracle BI Presentation Services Log Filters"](#)
- [Section 8.5.6, "Diagnosing Issues with Agents"](#)

For general information about logging in Oracle Business Intelligence, see [Section 8.3, "Understanding Diagnostic Log and Log Configuration Files."](#)

8.5.1 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 sawlogxx.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.

8.5.2 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.
Writer	disableCentralControl	(Optional) Determines that this entry is not updated by Fusion Middleware Control. Default value is true.

Table 8–3 (Cont.) Oracle BI Presentation Services Log Configuration File Elements

Element	Attribute	Description
Writer	implementation	<p>The following implementations are defined:</p> <ul style="list-style-type: none"> ■ FileLogWriter. Writes to a disk file. ■ OutWriter. 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. <p>Used in a production environment for information of some loggable but irrecoverable error (for example, failed NQTEST).</p> <p>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.</p> <p>On Windows, CrashWriter requires the appropriate version of <code>dbghelp.dll</code> (at least 6.0.17.0).</p> <p>The correct <code>dbghelp.dll</code> can be found in <code>support/windows/system32</code>.</p> <p>Put this DLL in the <code>WINNT/system32</code> or in the <code>main/bin</code> directory.</p> <p>No registration is required.</p>
Writer	name	Unique name for the writer.
Writer	writerClassId	<p>Specifies an integer number in the range 1 through 10. This number is used by filters to allow or prohibit logging.</p> <p>Each distinct writer must have a unique value, which is used later for filter configuration.</p> <p>Different writers might have the same class ID, but if they do, those writers cannot be distinguished by filters.</p>
Writer	fmtName	<p>(Optional) Specifies the format of logged messages. Valid values are:</p> <ul style="list-style-type: none"> ■ 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. <p>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.</p> <p>See Section 8.5.3, "Examples of the Formats of Logged Messages" for examples.</p>
Writer (FileLogWriter specific attribute)	dir	Specifies the directory where log files are created.

Table 8–3 (Cont.) Oracle BI Presentation Services Log Configuration File Elements

Element	Attribute	Description
Writer (FileLogWriter specific attribute)	ODLogFilePath	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 <code>bin/coredumps</code> 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: <code>saw.httpserver.request.soaprequest</code> 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.

8.5.3 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](#) shows the default format.

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](#) shows the ODL-TEXT format.

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: 1243446680218334471555761
]]

```

Example 8-7 shows the ODL-XML format.

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="124180446517874242628761" tid="127c">
<txt> Oracle BI Presentation Services has started successfully</txt>
<suppl_detail />
</msg>

```

8.5.4 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.
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, whereas 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.

8.5.5 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"/>
```

8.5.6 Diagnosing Issues with 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 8.3.2, "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 19.3.3.3, "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

In this file name:

- 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 information, see "Instance Properties in Job Manager" in *Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition*). Exit status shows the number of deliveries successfully completed.

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\coreapplicatio  
n_obisn\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 includes 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 quotation marks (") 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 quotation marks (") 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 a.m. on February 12, 2010, the file name 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.

Although 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 a 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

Table 9–1 (Cont.) Usage Tracking Output File Format

Column Number	Column Name	Data Type	Max Data Size	Nullable
2	Repository name	Varchar	128	No
3	Subject area name	Varchar	128	No
4	Node ID	Varchar	15	No
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 7 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.

Although 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 to a larger number 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.

Table 9–2 (Cont.) Usage Tracking Data

Column	Description
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.
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 Databases truncate 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.

Table 9–2 (Cont.) Usage Tracking Data

Column	Description
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.
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: <ul style="list-style-type: none"> ■ 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 nquery.log file, as described in Section 8.4.1.1, "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 includes the following chapters:

- [Chapter 10, "Configuring Repositories"](#)
- [Chapter 11, "Configuring Connections to External Systems"](#)
- [Chapter 12, "Configuring Presentation Setting Defaults"](#)
- [Chapter 13, "Configuring Mapping and Spatial Information"](#)
- [Chapter 14, "Configuring Time Zones"](#)
- [Chapter 15, "Localizing Oracle Business Intelligence"](#)
- [Chapter 16, "Configuring Currency Options"](#)
- [Chapter 17, "Configuring and Managing the Oracle BI Presentation Catalog"](#)

Configuring Repositories

You can use Fusion Middleware Control to upload the Oracle BI metadata repository and to set the location of the Oracle BI Presentation Catalog. This chapter includes the following topics:

- [Section 10.1, "Why Upload a Repository and Set the Catalog Location?"](#)
- [Section 10.2, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location"](#)
- [Section 10.3, "Using the System MBeans to Upload the Metadata Repository"](#)

10.1 Why Upload a Repository and Set the Catalog Location?

As the administrator, you must ensure that users can access the metadata repositories and Oracle BI Presentation Catalogs that contain the information that they require. The following list describes the importance of uploading the repository and setting the catalog location:

- **Upload Repository** — Uploading 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, you provide the name and location of the repository file and the current repository password. The uploaded repository is available to clients of Oracle BI Server, including to Oracle BI Presentation Services.

- **Set Catalog Location** — Setting the location of the Oracle BI Presentation Catalog ensures that its objects are available to users of Oracle Business Intelligence. The catalog is never uploaded. Instead, you copy the catalog files manually to either a local or shared location, then you set the location in Fusion Middleware Control.

10.1.1 Understanding Modes

You can upload the repository in either *shared mode*, which is recommended, or in *distributed mode*, as described in the following list:

- In shared mode, each Oracle BI Server instance loads the repository from the shared network location designated as the repository upload directory. You typically upload the repository in the recommended mode of *shared* when you have multiple Oracle BI Server components in the deployment. Repositories that are uploaded through Fusion Middleware Control are uploaded to both the shared location and the default local directory for each Oracle BI Server.

- 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.

Similarly to the metadata repository, the Oracle BI Presentation Catalog can be accessed in shared mode (which is recommended), or in distributed mode. Each Presentation Services component loads the Oracle BI Presentation Catalog from the catalog location that is specified in Fusion Middleware Control. In shared mode, changes to the catalog are propagated to all Presentation Services components. In distributed mode, if you make changes to a local catalog, then you must propagate the changes to all components.

10.1.2 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_
obisn\repository
```

If you have uploaded 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. This numbering ensures that the previous version is not overwritten. 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.

10.2 Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location

You can use Fusion Middleware Control to upload the metadata repository and to provide a shared location for the repository and a local or shared location for the Oracle BI Presentation Catalog.

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 uploaded repository (called the **Default RPD**). Note that the Data Source Name (DSN) visible to Oracle BI Server clients for the repository that is uploaded through Fusion Middleware Control is always **Star**.

3. Click **Lock and Edit Configuration** to allow changes to be made.

4. To upload 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 uses 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**.

As part of the Simple Installation type, Oracle Business Intelligence is configured with a sample default repository called SampleAppLite.rpd. See [Section 1.5, "Working with the Sample Application"](#) for more information.

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 about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "Introducing the Oracle BI Systems Management API."](#)

10.3 Using the System MBeans to Upload the Metadata Repository

The recommended approach for uploading the metadata repository is to use the Repository tab of the Deployment page, as described in [Section 10.2, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location."](#) You might encounter specific situations in which you must upload the repository using a different approach. For example, the size of the repository file can be too large for Fusion Middleware Control to handle, or you might be working in a development environment and have unique needs.

In such exception cases, you can use the System MBeans to manage repositories programmatically or from scripts, as described in [Chapter 22, "Introducing the Oracle BI Systems Management API."](#) You can also use the Fusion Middleware Control System MBean Browser to upload the default metadata repository, as described in the following procedure.

To use the System MBean Browser to upload the metadata repository:

1. Manually copy the repository file to upload to the local repository directory for each Oracle BI Server component in the deployment, at:


```
ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_obisn\repository
```
2. If the name or password of the repository has changed, then complete the following steps:
 - a. Use the System MBean Browser to specify the new name for the repository. See [Section 3.5, "Using the Fusion Middleware Control MBean Browser to Update Configuration Settings"](#) for more information.
 - b. Commit your changes.

- c. You must change the password for the repository file in the oracle.bi.enterprise credential map. Otherwise you cannot restart the BI Server, which requires this password to decrypt the file.
In Fusion Middleware Control, navigate to the bifoundation_domain node and from the menu, select **Security**, then select **Credentials**.
- d. Drill the oracle.bi.enterprise map, select the appropriate repository, and click **Create Key** to display the Create Key dialog in which you create a new credential.
- e. Create the key for the repository by completing the fields as follows:
 - **Key** = Set the key for the repository to `repository.name`, where *name* is the name of the repository, without the .rpd suffix.
 - **User Name** = Set the value to any name.
 - **Password** = Enter the password for the repository file, then retype it in the **Confirm Password** field.
- f. Click **OK**.

3. 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.

Configuring Connections to External Systems

This chapter explains how to configure connections to systems that are external to Oracle Business Intelligence. This chapter includes 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 Connections to Data Sources"](#)
- [Section 11.4, "Configuring for Connections to the Marketing Content Server"](#)
- [Section 11.5, "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 "Configuring a Database for the Oracle BI Scheduler" in *Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition*.

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 19, "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 19, "Configuring and Managing Agents"](#) for information about advanced configuring settings for agents.

For information about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "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 Connections to Data Sources

Connections to data sources are defined in the Oracle BI repository file (RPD). Repository developers use the Administration Tool to configure data source connections by importing metadata and configuring connection pools. See "Importing Metadata and Working with Data Sources" in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for more information.

You might need to update connection pool information in the repository during migrations to production and other environments. You can use the Oracle BI Server XML API to automate these connection pool changes. See "Moving from Test to

Production Environments" in *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition* for more information.

11.4 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 11.4.1, "Using Fusion Middleware Control to Configure the Connection to the Marketing Content Server"](#)
- [Section 11.4.2, "Manually Editing Settings for the Marketing Content Server Connection"](#)

11.4.1 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 about using methods in the Oracle BI Systems Management API to configure the connection to the content server, see [Chapter 22, "Introducing the Oracle BI Systems Management API."](#)

11.4.2 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 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.

11.5 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.

Although 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 about 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 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 "Managing Presentation Services Privileges" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

7. Restart Oracle BI Presentation Services to see the download option for Smart View.

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 18.4, "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 18, "Configuring and Managing Analyses and Dashboards"](#) for information about advanced configuring settings for analyses and dashboards.

For information about using methods in the Oracle BI Systems Management API to change configuration settings, see [Chapter 22, "Introducing the Oracle BI Systems Management API."](#)

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 includes the following topics:

- [Section 13.1, "What Are the System Requirements for Map Views?"](#)
- [Section 13.2, "Hardware Sizing and Deployment Strategy for Maps"](#)
- [Section 13.3, "Administering Maps"](#)

See [Chapter 20, "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, the system must include the following components:

- Oracle MapViewer, which is a J2EE service that serves as an engine for rendering maps using spatial data managed by Oracle Spatial. MapViewer is closely integrated with Oracle BI EE. MapViewer is installed as part of Oracle BI EE 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 20.1, "Configuring MapViewer to Support Map Views."](#)

- Spatial boundary data. NAVTEQ is one provider of 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 from the Oracle eLocation service. Terms and conditions of use are located at the following URL:
<http://elocation.oracle.com/elocation/legal.html>

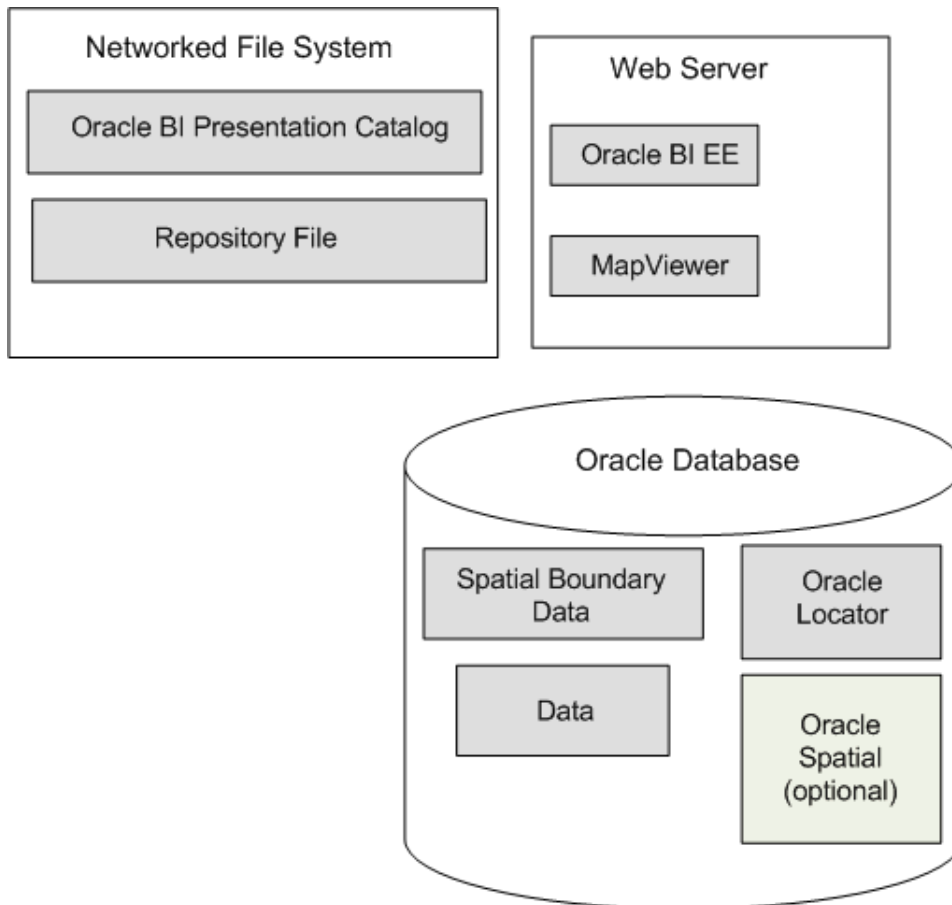
- Oracle Database, version 10g or later, to store the spatial data.

Oracle Locator, which is a feature of Oracle Database (all editions) that provides core location functionality needed by most customer applications.

If you use an Oracle Database as the Repository Creation Utility (RCU) database, then you can use that same Oracle Database for spatial data also. See *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence* for information.
- (Optional) 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.
- 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.

Figure 13–1 shows the default architecture for map views when Oracle BI EE is installed. You can store the data either in an Oracle Database or in other databases that Oracle BI EE supports. See Figure 20–1 for a diagram of the preferred architecture for map views.

Figure 13–1 Default Architecture for Map Views



When these pieces are in place, you administer the map using the Oracle BI Presentation Services Administration page, as described in [Section 13.3, "Administering Maps."](#)

13.2 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.

13.3 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 13.3.1, "Working with Maps and Layers"](#)
- [Section 13.3.2, "Administration Page Functions"](#)
- [Section 13.3.3, "Administering Maps Using Administration Pages"](#)
- [Section 13.3.4, "Handling the Translation of Layers in Maps"](#)

13.3.1 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 13.3.1.1, "Associating Layers with Columns"](#)
- [Section 13.3.1.2, "Ordering Layers on Maps"](#)
- [Section 13.3.1.3, "Changes to Spatial Metadata Require Restart"](#)

13.3.1.1 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.

See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for more information on double columns.

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.

13.3.1.2 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 the sort icon) before closing the dialog.

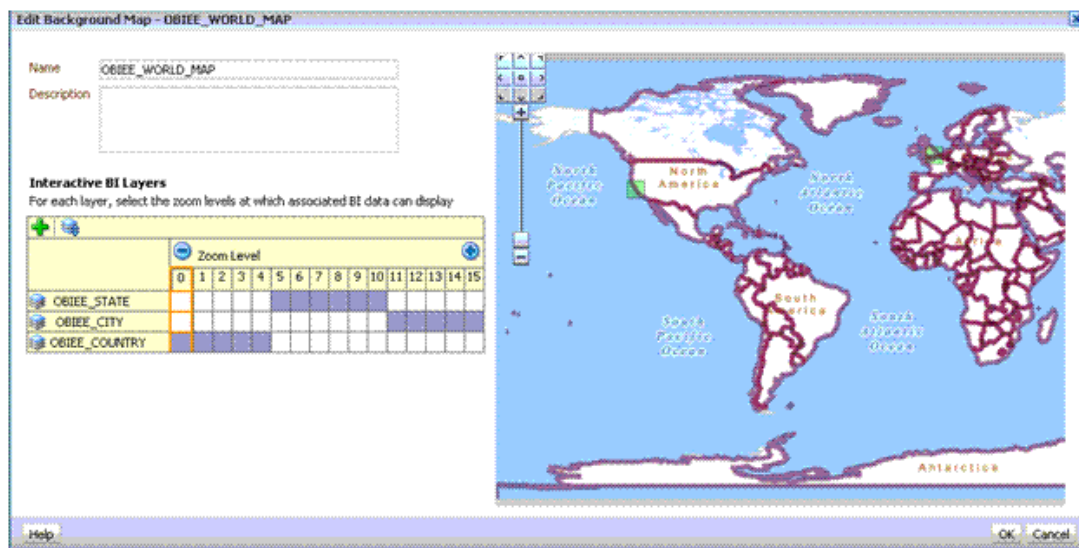
Layer ordering becomes irrelevant when the zoom ranges of layers do not intersect on the scale. Ordering becomes very important when layers have a common minimum and maximum zoom range. Use care to ensure that detailed layers are not hidden by the aggregated layers during drilling or zooming operations.

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 13–2 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 13–2 Edit Background Map Dialog with No Intersection

**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 13–3. Notice that all three layers include zoom level 4.

Figure 13–3 Edit Background Map Dialog with Intersection

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 about layers.

13.3.1.3 Changes to Spatial Metadata Require Restart

Administrators can edit the spatial metadata that is stored in the Oracle Database and accessed by MapViewer. For example, an administrator can add a new layer. These edits are not visible on the Oracle BI Presentation Services Administration pages for managing maps until MapViewer is restarted and so refreshed with the latest updates.

13.3.2 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 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.

13.3.3 Administering Maps Using Administration Pages

This section assumes that you installed Oracle BI Enterprise Edition and that Oracle MapViewer was automatically configured and deployed. For information on MapViewer, see *Oracle Fusion Middleware User's Guide for Oracle MapViewer*.

To administer maps using Administration pages:

1. Sign in to Oracle Business Intelligence.
2. In the global header, click **Administration**.
3. Click the **Manage Map Data** link to display the Manage Map Data page.
4. Click the Layers tab.
5. Click the **Import Layers** button to display the Import Layers dialog.
6. 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.

7. 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.

8. Click the Background Maps tab, then click the **Import Background Maps** button to display the Import Background Maps dialog.
9. 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.

10. 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.

11. 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.

12. In the dialog, select the connection and the images to use.

Click **OK** when you have finished selecting images.

13. 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.

13.3.4 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*.

Configuring Time Zones

This chapter includes the following topics on configuring time zones:

- [Section 14.1, "Why and Where are Time Zones Used?"](#)
- [Section 14.2, "Setting Time Zones"](#)
- [Section 14.3, "What is the Precedence Order for Time Zones"](#)
- [Section 14.4, "Where are Time Zone Specifications Stored?"](#)
- [Section 14.5, "Description of Time Zone Settings"](#)
- [Section 14.6, "Example: Configuration File Settings for Specifying the Time Zone"](#)

14.1 Why and Where are Time Zones Used?

Time zones are used throughout Oracle Business Intelligence for a variety of purposes. A time stamp can indicate when an object was changed, and users can specify a time for an agent to run. Users often are most comfortable working in their local time zones. As the administrator, you can configure the preferred time zones for users for various components.

Before you begin to set preferred time zones, see [Table 14–1](#) for information about where time zones are used.

Table 14–1 Time Zone Usage

Type	Description
Oracle BI Presentation Services	<p>If you have users in time zones that are different from the zone for Presentation Services, then you as the administrator 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.</p> <p>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.</p> <p>For information about how users specify their preferred time zones, see <i>Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition</i>.</p>

Table 14–1 (Cont.) Time Zone Usage

Type	Description
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.
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: <ul style="list-style-type: none"> ▪ 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.

14.2 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 14–2](#) for information about the precedence order for time zones.
 - See [Table 14–3](#) for descriptions of the session variables and elements.
 - See [Section 14.6, "Example: Configuration File Settings for Specifying the Time Zone."](#)
 - See *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for complete information about 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.

14.3 What is the Precedence Order for Time Zones

The actual time zone in which various types of content are displayed follows a precedence order that [Table 14–2](#) describes. In the table, the items with lower numbers override those with higher numbers. For example, Item 1 takes precedence over Item 2.

Table 14–2 Precedence Order for Time Zones

Time Zone For	Determined By
Data	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 1. "User-Preferred Time Zone" 2. The time zone for Oracle BI Presentation Services.
Log file information	<ol style="list-style-type: none"> 1. The setting of the Logging element in the instanceconfig.xml file. 2. The time zone for Presentation Services.

14.3.1 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.

14.4 Where are Time Zone Specifications 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 delete those zones that users in your organization does not use.

14.4.1 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 & Canada); Tijuana"

14.5 Description of Time Zone Settings

Table 14–3 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 14–3 Time Zone Settings

Element	Session Variable	Description	Value
DefaultDataOffset	DATA_TZ	<p>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.</p> <p>If you do not set this option, then no time zone conversion occurs because the value is "unknown".</p> <p>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.</p>	<p>An offset that indicates the number of hours away from GMT time. For example: "GMT-05:00" or "-300", which means minus 5 hours.</p>
DefaultDataDisplay	DATA_DISPLAY_TZ	<p>Specifies the time zone to use for displaying data.</p> <p>If you do not set this option, then the value is the "User-Preferred Time Zone".</p>	<p>One of the time zones that are specified in the TimeZones.xml file.</p> <p>See Section 14.4.1, "Specifying Time Zone Values."</p>

Table 14-3 (Cont.) Time Zone Settings

Element	Session Variable	Description	Value
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 14.4.1, "Specifying Time Zone Values."
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 14.4.1, "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

14.6 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 & Canada); Tijuana</Logging>
  <DefaultUserPreferred>(GMT-08:00) Pacific Time (US & Canada);
  Tijuana</DefaultUserPreferred>
  <DefaultDataDisplay>(GMT-06:00) Central Time (US &
  Canada)</DefaultDataDisplay>
</TimeZone>
```

See [Section 3.4, "Using a Text Editor to Update Configuration Settings"](#) for information about editing the instanceconfig.xml file.

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 about how to configure for deployment in one or more language environments other than English. This chapter includes the following topics:

- [Section 15.1, "What Is Localization?"](#)
- [Section 15.2, "Localizing Oracle BI Presentation Services"](#)
- [Section 15.3, "Setting the Current Locale in Catalog Manager"](#)
- [Section 15.4, "Setting the Current Locale in the Oracle BI Server"](#)
- [Section 15.5, "Localizing Metadata Names in the Repository"](#)
- [Section 15.6, "Supporting Multilingual Data"](#)

15.1 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 about supported languages, see [Section 1.8, "System Requirements and Certification."](#)

15.1.1 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.

15.1.2 Tasks for Localizing 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 15.2, "Localizing Oracle BI Presentation Services"](#)
- [Section 15.4, "Setting the Current Locale in the Oracle BI Server"](#)

- [Section 15.5, "Localizing Metadata Names in the Repository"](#)
- [Section 15.6, "Supporting Multilingual Data"](#)

15.2 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 15.2.1, "Localizing the User Interface for Oracle BI Presentation Services"](#)
- [Section 15.2.2, "Localizing Oracle BI Presentation Catalog Captions"](#)
- [Section 15.2.3, "Tip for Arabic and Hebrew in Mozilla Firefox Browsers"](#)

15.2.1 Localizing the User Interface for Oracle BI Presentation Services

You can localize the user interface for Oracle BI Presentation Services, 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 15.2.1.1, "Understanding the Directory Structure for Localizing Presentation Services"](#)
- [Section 15.2.1.2, "Localizing Messages for Users' Preferred Currency"](#)
- [Section 15.2.1.3, "Specifying the Default Language for the Sign-In Page"](#)
- [Section 15.2.1.4, "Configuring the Languages and Locales for the Sign-In Page"](#)
- [Section 15.2.1.5, "Specifying the Language in the URL"](#)

15.2.1.1 Understanding the Directory Structure for Localizing 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_obipsn\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.

15.2.1.2 Localizing Messages for Users' Preferred Currency

Section 16.2.1, "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\coreapplicati
on_obipsn\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`.

15.2.1.3 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.

15.2.1.4 Configuring the Languages and Locales for the Sign-In Page

You can configure the languages and locales that are available to users on the sign-in page. This ability is helpful for limiting the number of languages and locales that users can access. You use the AllowedLanguages and AllowedLocales elements in the instanceconfig.xml file to specify the available languages and locales.

To manually configure the languages and locales that are available:

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 following elements:
 - AllowedLanguages — Specifies the languages that are available for selection, as a comma-delimited list. You can specify a list of the following identifiers, which are ISO 639 language codes:

ar
 es
 da
 de
 el
 en
 es
 fi
 fr
 he
 hr
 hu
 it
 ja
 ko
 nl
 no
 pl
 pt
 pt-BR
 ro
 ru
 sk
 sv

th
tr
zh-CN
zh-TW

- AllowedLocales — Specifies the locales that are available for selection, as a comma-delimited list. You can specify any definition from the localeDefinitions.xml file in the ORACLE_HOME/bifoundation/web/display directory. You specify the locales using ISO 639 language codes followed by ISO 3166 country codes. Examples include fr-fr and fr-ca.
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <Localization>
    <AllowedLanguages>en, fr, pt-BR</AllowedLanguages>
    <AllowedLocales>en-ca, en-us</AllowedLocales>
  </Localization>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

15.2.1.5 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 15–1 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 15–1 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

Table 15–1 (Cont.) Example of Settings in the Symbolic URL Parameters Applet

Name	Type	Path Argument Value	Append	Sequence #
PostRequest	Command	PostRequest	Y	5
Lang	Profile Attribute	LanguageCode	Y	6

15.2.2 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.

This section describes the steps in the process of localizing captions:

- [Section 15.2.2.1, "Step 1: Understanding the Export Process"](#)
- [Section 15.2.2.2, "Step 2: Exporting Text Strings in the Catalog"](#)
- [Section 15.2.2.3, "Step 3: Editing Exported Strings in XML Files"](#)
- [Section 15.2.2.4, "Step 4: Handling Duplicate Exported Text Strings"](#)
- [Section 15.2.2.5, "Step 5: Exposing Text Strings in the Catalog"](#)

15.2.2.1 Step 1: Understanding the Export Process

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.

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_
obipsn\msgdb\l_xx\captions
```

The export process not only generates new XML files, but the process also modifies the catalog, inserting the appropriate message ID for each object. Presentation Services uses those message IDs to locate the newly translated text.

15.2.2.2 Step 2: Exporting Text Strings in the Catalog

The following procedure describes how to export text strings in the catalog.

To export text strings in the catalog:

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 offline mode, because you might lack permission to modify all objects in online 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 analysis that is shown in [Figure 15–1](#) can be exported for translation.

Figure 15–1 Title in an Analysis for Export and Translation

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	Modem	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.

15.2.2.3 Step 3: Editing Exported Strings in XML Files

When the export process is complete, deliver the output XML file to the localization team.

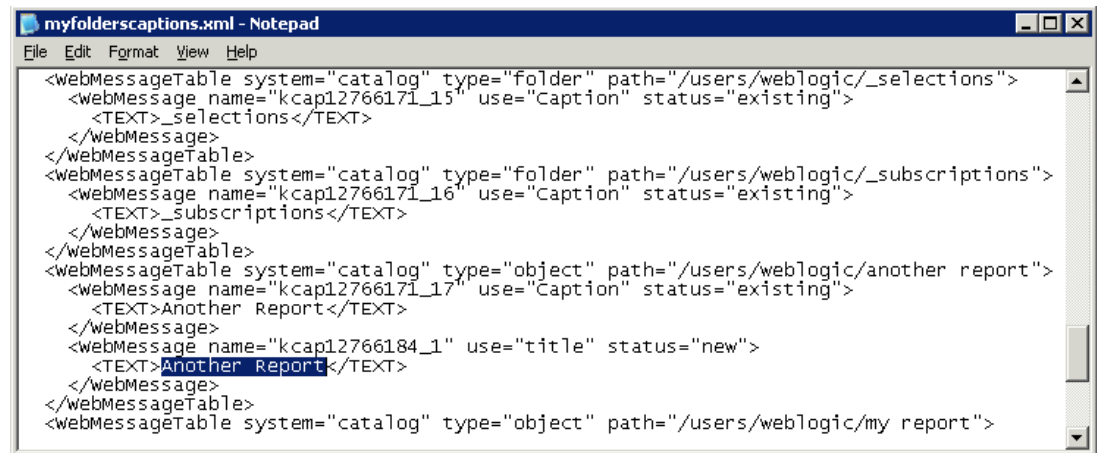
When editing XML files, use an editor that is designed for XML files. Ensure that you follow the encoding that is specified at the top of the XML file and that you escape

special characters as appropriate. You and the localization team are responsible for resolving any errors in the translated text strings. Consider that the contents of the catalog are updated whenever objects are added, deleted, or modified.

You can make a copy of every output file for each language to be translated.

Figure 15–2 shows an extract from an exported caption XML file before translation. The file is named `myfoldercaptions.xml`. Figure 15–3 shows an extract from the file after translation. The file is named `myfoldercaptions_fr.xml`.

Figure 15–2 Caption XML File Before Translation

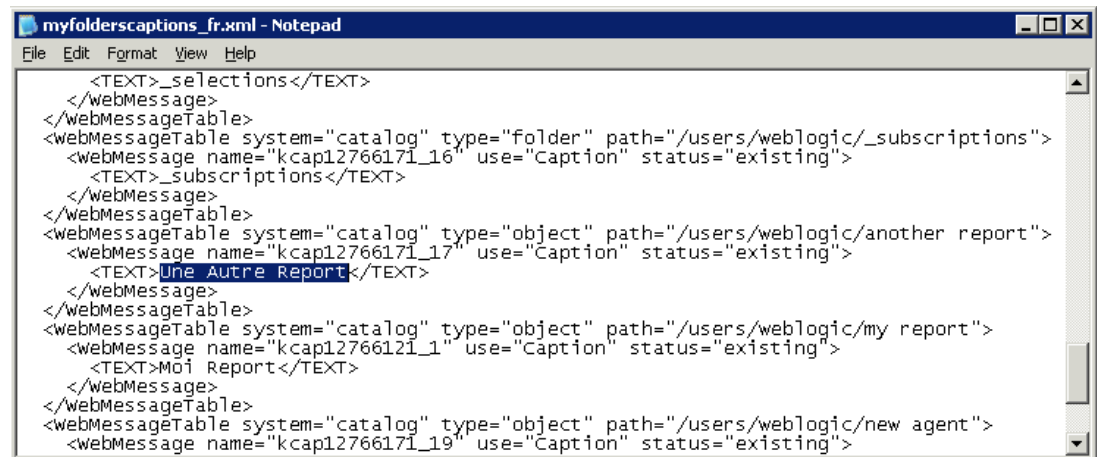


```

myfoldercaptions.xml - Notepad
File Edit Format View Help
<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">

```

Figure 15–3 Caption XML File After Translation



```

myfoldercaptions_fr.xml - Notepad
File Edit Format View Help
  <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">

```

15.2.2.4 Step 4: Handling Duplicate Exported Text Strings

You might encounter an issue of having duplicate exported text strings from the catalog. This situation arises when the Export Captions utility is run simultaneously by multiple users or if the same user runs the utility twice in less than one minute. The following procedure describes how to address duplicate captions.

To handle duplicate exported text strings:

1. Run the Export Captions utility, as described in [Section 15.2.2.2, "Step 2: Exporting Text Strings in the Catalog."](#)

2. In Catalog Manager, with the catalog still open in offline mode, select the folder that contains the strings to export.
3. From the **Tools** menu, select **Export Captions**.
4. Click **Browse** to select the location in which to write the output file, then click **OK**.
5. In the "What to do with duplicate captions" section, select one of the following options:
 - **Leave alone** — Specifies to leave the duplicate captions as they are.
 - **Remove IDs** — Specifies to generate new and unique IDs for any objects that contain duplicate or invalid IDs.
 - **Remove texts** — Specifies to generate new and unique IDs for any objects that contain invalid IDs and to delete the text string for an objects that contain valid but duplicate IDs.

This option results in two objects with the same name being included in directory listings of the catalog. This option is most useful for a set of analyses that have unique names and message IDs, but that share the same text strings for components such as graph axes and legends. This option results in needing to translate components only once. For example, you might translate "X Axis" only once rather than once for each graph.

6. Click **OK**.

Consider the following webmessages.xml file, which contains duplicate captions:

```
<WebMessageTable system="catalog" type="folder" path="/shared/example/A">
  <WebMessage name="kcap12790830_5" use="Caption" status="new">
    <TEXT>A Really Good Report</TEXT>
  </WebMessage>
</WebMessageTable>
<WebMessageTable system="catalog" type="folder" path="/shared/example/B">
  <WebMessage name="kcap12790830_5" use="Caption" status="new">
    <TEXT>I like this report</TEXT>
  </WebMessage>
</WebMessageTable>
<WebMessageTable system="catalog" type="folder" path="/shared/example/Copy of A">
  <WebMessage name="kcap12790830_5" use="Caption" status="new">
    <TEXT>A Really Good Report</TEXT>
  </WebMessage>
</WebMessageTable>
```

In this example file, Object B has an invalid duplicate message ID. Object Copy of A has a valid but duplicate message ID. You can make the following selections in the Export Captions dialog:

- Selecting **Leave alone** makes no changes to the contents of the file.
- Selecting **Remove IDs** generates new and unique IDs for both Object B and Object Copy of A.
- Selecting **Remove texts** generates a new and unique ID for Object B and deletes the WebMessage element for Object Copy of A. While this option generally ensures fewer messages to translate, keep in mind that you now see two objects with the same name in a directory listing of the catalog in Presentation Services and in Catalog Manager.

15.2.2.5 Step 5: Exposing Text Strings in the Catalog

After you have exported the text strings for the catalog, you must expose them for users.

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\coreapplicati  
on_obipsn\msgdb\l_xx\captions
```

where *xx* is the language extension.

For example:

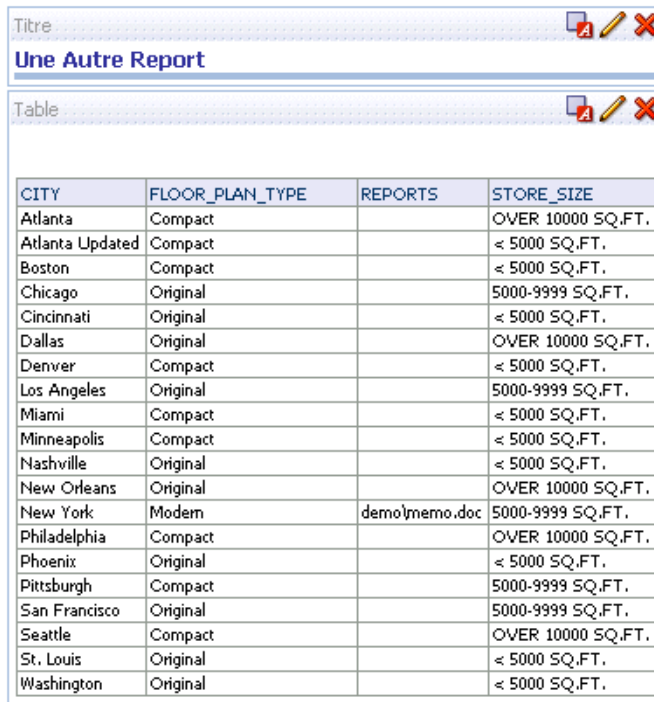
```
D:\R11\instances\instance1\bifoundation\OracleBIPresentationServicesCompon  
ent\coreapplication_obips1\msgdb\l_fr\captions\myfoldercaptions_fr.xml
```

Other examples of language extensions include *cs* for Czech and *de* for German.

Note: The *ORACLE_*
INSTANCE\bifoundation\OracleBIPresentationServicesComponent\c
oreapplication_obipsn\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.

2. The export process not only generates new XML files, but the process also modifies the catalog, inserting the appropriate message ID for each object. After placing the translated XML files in the directory as specified in Step 1, place the modified catalog in either its default location or in the appropriate location that the system now uses. See [Section 10.2, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location"](#) for information.
3. Sign into Oracle Business Intelligence and select the appropriate language, such as French.
4. Display the translated content.

For example, display the translated title in an analysis, as shown in [Figure 15-4](#).

Figure 15–4 Title in an Analysis After Export and Translation


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.

15.2.3 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 about changing the `layout.scrollbar.side` setting, see the Firefox documentation.

15.3 Setting the Current Locale in Catalog Manager

When you use Catalog Manager, you can specify the locale to use for its user interface elements and for objects in the catalog. These locales can be the same or different. The user interface elements are available in 10 locales, and the catalog content for certain applications is available in 28 locales.

You can see the user interface elements of Catalog Manager (dialogs, menus, and so on) in any of the 10 locales in which it is available. Certain areas of Catalog Manager,

such as data handling, are not yet translated or localized. Catalog Manager uses the following process to decide which locale to display:

1. Check for the setting of the "-nl <locale>" parameter when Catalog Manager is started. You set this parameter as part of the CATMAN_VMARGS variable in the runcat.cmd or runcat.sh file, as shown in the following examples:

```
set CATMAN_VMARGS=-nl fr -vmargs -Xmx1024M -Dosgi.clean=true
set CATMAN_VMARGS=-nl fr_CA -vmargs -Xmx1024M
-Dosgi.clean=true
```

2. Check for the default locale for Java, as specified on your computer.
3. Using the default locale of English (specifically, en_US).

When you start Catalog Manager and open a catalog in online mode, you can select the locale for viewing the contents of the catalog. The locales that are available for selection depend on the following criteria. Catalog Manager uses this selection for subsequent online connections.

- Whether that locale was selected for Presentation Services during the installation process.
- Whether the contents of the catalog have been translated for a specified locale.

If translated files are not available for that locale, then the contents are displayed in the default locale of English (specifically, en_US).

For more information on Catalog Manager, see [Section 17.3, "About Catalog Manager."](#)

15.4 Setting the Current Locale in the Oracle BI Server

The following sections provide information about setting the locale in Oracle BI Server:

- [Section 15.4.1, "Setting Locale Parameters on the Oracle BI Server"](#)
- [Section 15.4.2, "Understanding How the Error Message Language Is Determined"](#)
- [Section 15.4.3, "Troubleshooting the Current Locale in the Oracle BI Server"](#)

15.4.1 Setting Locale Parameters on the 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 about 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.

15.4.1.1 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 15-2](#).

[Table 15-2](#) 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 15-2 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

Table 15–2 (Cont.) LOCALE Settings for UNIX Platforms

Locale (Platform-Independent Name)	Name on Solaris	Name on AIX	Name on HP-UX/Linux
Romanian	ro_RO.UTF-8	RO_RO.UTF-8	ro_RO.utf8
Russian	ru_RU.UTF-8	RU_RU.UTF-8	ru_RU.utf8
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

15.4.2 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. Therefore, these clients follow the precedence model starting with the ODBC DSN error message setting.

15.4.3 Troubleshooting the Current Locale in the Oracle BI Server

This section provides the following information about troubleshooting the current locale in the Oracle BI Server:

- [Section 15.4.3.1, "Handling the NLS Locale Not Supported Error Message"](#)
- [Section 15.4.3.2, "Setting the Japanese Locale on AIX Systems"](#)

15.4.3.1 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 NQSSConfig.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 15-2](#) for the requested language.
- **Windows.** Add the corresponding language pack using the Regional Settings dialog.

15.4.3.2 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 NQSSConfig.INI file for editing, as described in [Section 3.4, "Using a Text Editor to Update 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 NQSSConfig.INI file.

15.5 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 *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for more information.

15.6 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 15.6.1, "What is Multilingual Data Support?"](#)
- [Section 15.6.2, "What is Lookup?"](#)
- [Section 15.6.3, "What is Double Column Support?"](#)
- [Section 15.6.4, "Designing Translation Lookup Tables in a Multilingual Schema"](#)
- [Section 15.6.5, "Creating Logical Lookup Tables and Logical Lookup Columns"](#)
- [Section 15.6.6, "Creating Physical Lookup Tables and Physical Lookup Columns"](#)
- [Section 15.6.7, "Supporting Multilingual Data in Essbase Through Alias Tables"](#)
- [Section 15.6.8, "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*.

15.6.1 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.

15.6.2 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.

15.6.3 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.

15.6.4 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 15.6.4.1, "A Lookup Table for Each Base Table"](#)
- [Section 15.6.4.2, "A Lookup Table for Each Translated Field"](#)

15.6.4.1 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 15-5](#) shows each record in the lookup table matching only one row in the base table.

Figure 15–5 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

15.6.4.2 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, as shown in [Figure 15–6](#). 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 15–6 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

15.6.5 Creating Logical Lookup Tables and Logical Lookup Columns

This section describes creating logical lookup tables and lookup columns and contains the following topics:

- [Section 15.6.5.1, "Creating Logical Lookup Tables"](#)
- [Section 15.6.5.2, "Designating a Logical Table as a Lookup Table"](#)
- [Section 15.6.5.3, "About the LOOKUP Function Syntax"](#)
- [Section 15.6.5.4, "Creating Logical Lookup Columns"](#)

15.6.5.1 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 15.6.5.2](#),

"Designating a Logical Table as a Lookup Table." Figure 15–7 provides an example of a lookup table.

Figure 15–7 Example of a Lookup 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 15–7, 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 15–7, 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.

15.6.5.2 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.

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.

15.6.5.3 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")

15.6.5.4 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 preceding example, 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 preexisting 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 15–3](#) provides a mapping table for non-ISO language codes. LANGUAGE is a non-ISO language code.

Table 15–3 Mapping Table for Non-ISO Language Codes

WEBLANGUAGE	LANGUAGE	LAN_INT
en	ENG	0
cn	CHI	1
fr	FRA	1

15.6.6 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, 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 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.
9. Click **OK** in the Logical Column dialog.
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 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 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 for the physical translation table. For example, on the Keys tab of the Physical Table dialog 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 [Section 15.6.5, "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.

15.6.7 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.

15.6.8 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 perform this configuration, ensure that ORDERBY_SUPPORTED is not selected in the Features tab of the Database dialog in the Administration Tool. See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about 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.

Configuring Currency Options

When content designers create analyses, they often include data that shows currency, such as American dollars. As the administrator, you can perform various tasks that affect currency options that are available to users. This chapter includes the following topics:

- [Section 16.1, "Changing the Default Currency for Analyses"](#)
- [Section 16.2, "Defining User-Preferred Currency Options"](#)

16.1 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-1" 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-1"` 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.

16.2 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 16–1 Example of the Currency Box on the My Account Dialog: Preferences Tab

The screenshot shows the 'My Account' dialog box with the 'Preferences' tab selected. The 'Currency' dropdown menu is set to 'Euro'. Other settings include 'Starting Page' (My Dashboard), 'Locale (location)' (Default - English - United States), 'User Interface Language' (Default - English), 'Time Zone' (Default - Unknown Time Zone), and 'Accessibility Mode' (Off).

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	<input type="radio"/> On <input checked="" type="radio"/> 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 so that each user sees a list specific to his or her account. Each user sees a subset of the total options available. For example, one group of users might want to select from only Global Currency 1 or Global Currency 2 while another group might want to select from different options.

Note: 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 "Configuring Logical Columns for Multicurrency Support" in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

16.2.1 Defining User-Preferred Currency Options Using a Static Mapping

You can use a mapping to define a static list of options that all users see for selecting currency.

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_
obip$
```

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" />
```

In this format:

- 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 15.2.1.2, "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.

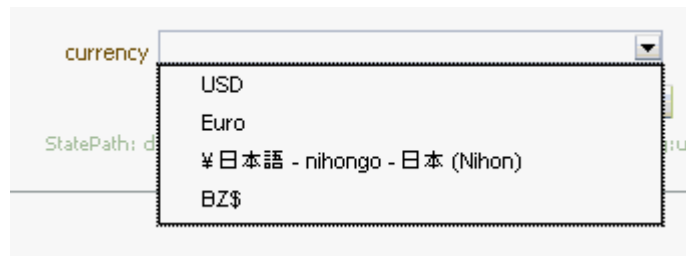
For information, see [Section 4.1, "About Starting and Stopping Oracle Business Intelligence."](#)

16.2.2 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="gc1" displayText="Global Currency 1"
    currencyTag="int:USD" />
  <UserCurrencyPreference sessionVarValue="gc2" displayText="Global Currency 2"
    currencyTag="int:euro-1" />
  <UserCurrencyPreference sessionVarValue="gc3" displayText="Global Currency 3"
    currencyTag="loc:ja-JP" />
  <UserCurrencyPreference sessionVarValue="orgc" displayText="Org Currency"
    currencyTag="loc:en-BZ" />
</UserCurrencyPreferences>
```

[Figure 16–2](#) shows how these values from the userpref_currencies.xml file are displayed in a drop-down list of currency options for a prompt on a dashboard page. The drop-down list is similar to what is displayed for the **Currency** box on the My Account dialog: Preferences tab.

Figure 16–2 Static Currency Options in a Prompt

16.2.3 Defining User-Preferred Currency Options Using a Dynamic Mapping

You can use a mapping to define a dynamic list of options that users see for selecting currency. The list 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.

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_
obip$
```

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>
```

In this format:

- *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."](#)

16.2.4 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 16–1](#) shows sample results from the logical SQL statement.

Table 16–1 Sample Logical SQL Results

"Markets"."UserPreference"	"Markets"."CurrencyTag"	"Markets"."UserPreferenceName"
varchar	varchar	varchar
orgc1	loc:en-BZ	Org currency
gc2	int:euro-1	Global currency 2
lc1	int:DEM	Ledger currency
gc1	int:USD	Global Currency 1

[Figure 16–3](#) shows how the values that are generated dynamically from the SQL statement in the `userpref_currencies.xml` file are displayed in a drop-down list of currency options for the **Currency** box on the Preferences tab of the My Account dialog. The drop-down list is similar to what is displayed for a prompt on a dashboard page.

Figure 16–3 Dynamic Currency Options in the My Account Dialog

The screenshot shows the 'My Account' dialog box with the following details:

- User ID:** Administrator
- Display Name:** Administrator
- Navigation Tabs:** Preferences (selected), BI Publisher Preferences, Delivery Options, Roles and Catalog Groups
- Starting Page:** My Dashboard
- Locale (location):** Default - English - United States
- User Interface Language:** Default - English
- Time Zone:** Default - Unknown Time Zone
- Currency:** BZ\$ (dropdown menu is open showing options: BZ\$, Euro, DEM, USD)
- Accessibility Mode:** Off

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 includes the following topics:

- Section 17.1, "About the Oracle BI Presentation Catalog"
- Section 17.2, "Maintaining the Oracle BI Presentation Catalog"
- Section 17.3, "About Catalog Manager"
- Section 17.4, "Starting Catalog Manager and Opening Catalogs"
- Section 17.5, "Using the Catalog Manager Workspace"
- Section 17.6, "Working with Objects in Catalog Manager"
- Section 17.7, "Viewing and Editing Catalog Objects in XML"
- Section 17.8, "Searching for and Replacing Catalog Text Using Catalog Manager"
- Section 17.9, "Creating Reports to Display Catalog Data Using Catalog Manager"
- Section 17.10, "Archiving and Unarchiving Using Catalog Manager"
- Section 17.11, "Configuring for Full-Text Catalog Search"
- Section 17.12, "Replicating Oracle BI Presentation Catalogs"

17.1 About the Oracle BI Presentation Catalog

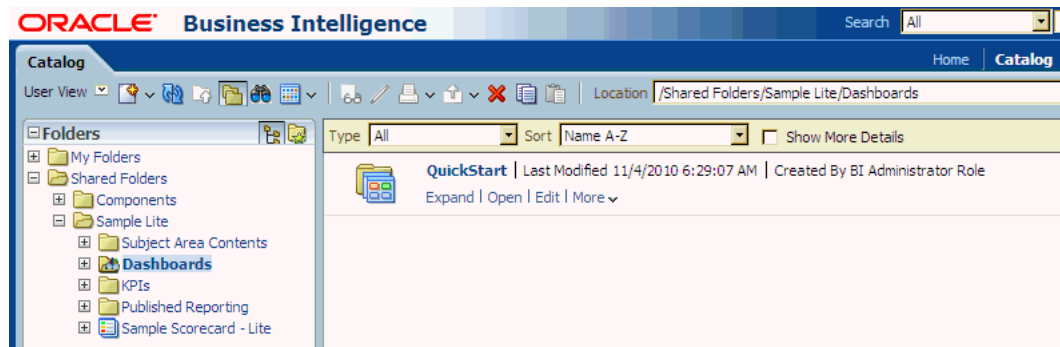
The Oracle BI Presentation Catalog stores the content that users create in a directory structure of individual files. This content includes folders, shortcuts, Oracle BI EE objects (such as analyses, filters, prompts, and dashboards), and Oracle BI Publisher objects (such as reports and templates).

This section contains the following topics:

- Section 17.1.1, "Objects in the Catalog"
- Section 17.1.2, "Locations for the Catalog"
- Section 17.1.3, "File System Guidelines for Catalogs"

17.1.1 Objects in the Catalog

Figure 17–1 shows sample objects in the catalog, as seen in Presentation Services.

Figure 17–1 Sample Objects in the Catalog in Presentation Services

17.1.1.1 Guidelines for Object Names

Each object in the catalog is stored in its own file. For example, an analysis called Analysis 1 is stored in a file named Analysis1. The object name that is visible to users, such as Analysis 1, is referred to as the logical object name.

The following list provides guidelines for object names:

- No restrictions exist on which characters are allowed in the logical name of an object in the catalog, provided that the characters are valid Unicode characters. The following are valid logical names:
 - Hello World
 - Profit / Loss
 - % Sales * \$ Cost ~~ \$ "Expense"?
- The length of the logical object name must not exceed 256 Unicode characters.
- The length of the logical path name for an object must not exceed 16000 Unicode characters.
- The number of directory segments in a logical path name for an object must be not exceed 255 segments.

For example, a directory with a name such as
 /n1/n2/n3/n4/.../n253/n254/n255 is acceptable, while a name such as
 /n1/n2/n3/n4/.../n254/n255/n256 is unacceptable.

- When you pass the path name of an object using SOAP, you must escape the following characters:

```
Forward slash (/)
Backward slash (\)
Tilde (~)
Asterisk (*)
Question mark (?)
```

The following logical path names are all valid:

```
/shared/test/Hello World
/shared/test/Profit \ / Loss
/shared/test/% Sales \* $ Cost \~\~ $ "Expense"?
```

Use care when building a catalog path. It is very common to see code that assumes the forward slash (/) is always a path separator. Always verify your path code with an object name such as "Profit / Loss".

- When you pass a catalog search filter using SOAP, you must escape the following characters:

```
Forward slash (/)
Backward slash (\)
Tilde (~)
Asterisk (*)
Question mark (?)
Caret (^)
Dollar sign (?)
```

The following search filters are all valid:

```
Hello World
Profit \ / Loss
% Sales \* \$ Cost \~\~ \$ "Expense" \?
```

17.1.1.2 Attribute Files for Objects

Each object has a corresponding attribute file. For example, the analysis called Analysis1 would have a corresponding attribute file named Analysis1.atr. The attribute 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.

17.1.1.3 Lock Files for Objects

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 catalog might not be removed completely. If Presentation Services reports of such a lock file, then you must delete it manually.

17.1.2 Locations for the Catalog

The following list provides the default locations for the directory that holds catalog files:

- On Windows systems:

```
ORACLE_
INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplicati
on_obipsn\catalog
```

- On UNIX systems:

```
ORACLE_
INSTANCE/bifoundation/OracleBIPresentationServicesComponent/coreapplicati
on_obipsn/catalog
```

You are not required to store the catalog in its default location. You can store the catalog in other locations. When you work in a cluster configuration, you store the catalog on a shared file system that all nodes in the cluster can access. For information, see [Section 5.4.2, "Sharing the Oracle BI Presentation Catalog."](#)

See [Section 17.1.3.3, "Handling Catalog Files on Various Platforms"](#) for more information on catalog directory names.

17.1.3 File System Guidelines for Catalogs

This section describes the following guidelines for working with objects in catalogs in file systems:

- [Section 17.1.3.1, "Handling Users of the Catalog"](#)
- [Section 17.1.3.2, "Handling Heterogeneous Nodes"](#)
- [Section 17.1.3.3, "Handling Catalog Files on Various Platforms"](#)
- [Section 17.1.3.4, "Known Issues with Catalog Files"](#)

17.1.3.1 Handling Users of the Catalog

The catalog is designed to scale to thousands of concurrent users. To achieve this scaling, the catalog adheres to the following guidelines:

- The average user typically only reads from the catalog and rarely, if ever, writes to it. In Release 11g, each user is constantly and automatically updating his or her Most Recently Used file, but each user's "read" operations still far outweigh the user's "writes" operations. Therefore, the read-to-write ratio is typically at least 100 to 1.
- While a locking mechanism guarantees that only one user can write to an object at a time, it is rare for multiple users to attempt to write simultaneously to the same object. A feature called "lazy locking" allows users to continue reading an object even when another user is updating that object.
- Modern file systems cache "small" files directly inside the directory record, such that reading any information on a directory simultaneously loads all small files directly into the operating system's memory cache. Therefore, it is good practice to keep files in the catalog "small," especially the frequently "read" .atr metadata files. When these metadata files remain small, then all the .atr files in a directory are loaded into memory with one physical hard disk read. Every file that exceeds the "small" threshold adds another physical hard disk read, which can cause a 100% degradation for each large file. In other words, use care when considering storing arbitrary "Properties" in .atr files.
- Reading an object's .atr metadata file using NFS is far slower than reading it directly from a local disk. For this reason, Presentation Services additionally caches all .atr files internally. This cache can become briefly "stale" when another node in the cluster writes data to the file that is newer than the data that is cached by the current node. Therefore, all nodes are refreshed according to the MaxAgeMinutes element in the instanceconfig.xml, whose default for a cluster is 5 minutes. This default setting commonly achieves the best trade-off between the possibility of stale data and the known performance impact. (The default for an environment without clusters is 60 minutes.)

You can modify the MaxAgeMinutes element for your system. Its parent elements are Cache and CatalogAttributes. Before you modify the element, ensure that you are familiar with the information in [Section 3.4, "Using a Text Editor to Update Configuration Settings."](#)

17.1.3.2 Handling Heterogeneous Nodes

To allow heterogeneous nodes in a cluster, the catalog adheres to the following guidelines:

- The maximum length for the name of an object on disk is 256 bytes, which is 64 Unicode characters. The logical name is restricted to 256 Unicode characters. To adhere to this restriction, logical names greater than 32 characters are hashed.

- The maximum length for the name of a path on disk is 32KB, which is 8000 Unicode characters. The logical path is restricted to 16000 Unicode characters.
- All path names on disk are all lowercase. The logical path name allows mixed case, but is still case-insensitive.
- Certain characters are not allowed for path names on disk, while the logical path name allows all characters. For example, Windows systems disallow certain characters such as the colon (:), so those characters are mapped using standard HTML escape sequences. For example, the period character (.) becomes "%2e".
- Certain file names are not allowed on disk, while the logical object name has no restrictions. For example, Windows systems disallow certain file names such as COM, so those names are mapped using standard HTML escape sequences. For example, "com" becomes "co%6d".

17.1.3.3 Handling Catalog Files on Various Platforms

Keep the following points in mind when handling catalog files on various platforms:

- For UNIX Platforms: UNIX kernels must commonly be configured to allow more than 4000 subdirectories per directory. See [Section 17.2.1, "Manually Changing Additional Configuration Settings for the Catalog"](#) for information on the HashUserHomeDirectories element.

- For Windows Platforms:

When users want to navigate catalog files using a tool such as Microsoft Windows Explorer, then they want the catalog structure based on a short path name such as c:/obi/demo, rather than the long default path name. Note that such navigation is not recommended.

- FAT is not supported, and NTFS is required.
- Performance on Windows platforms degrades noticeably when more than 8000 files exist in a single directory. Because each catalog object has two files (the data file and the .atr metadata file), it is strongly recommended that you not store more than 4000 catalog objects in a single directory. See [Section 17.2.1, "Manually Changing Additional Configuration Settings for the Catalog"](#) for information on the HashUserHomeDirectories element.
- Windows Explorer does not handle long path names properly, and it is recommended to not use Windows Explorer to navigate the internal structure of the catalog. While the file system can handle path names as large as 32KB and Presentation Services is not negatively affected, you cannot use Windows Explorer with any path name that is longer than approximately 2KB.

Because a single Unicode character can require as many as 4 bytes, you might be unable to use Windows Explorer with path names of only 500 Unicode characters. This limitation does not affect Presentation Services. Because of this limitation, place the catalog in a top-level directory, such as c:\mycatalog\sales.

17.1.3.4 Known Issues with Catalog Files

The following issues are known when working with catalog files:

- Locking across NFS systems is difficult, but Presentation Services provides an effective locking mechanism in recent patches. Obtain key patches to update older versions of Oracle BI EE as necessary.

For more information, see [Section 17.2.4, "Validating the Catalog."](#)

- Various third-party FTP programs have issues handling '%' escape sequences, which often results in a renamed file that is doubly escaped. For example, a file that is named sa%2epaint (whose logical name is SA.Paint) is incorrectly renamed to sa%252epaint (whose logical name is SA%2ePaint).

Avoid using an FTP program directly against a catalog. Instead, download and use the 7-Zip utility to compress the catalog files, then use an FTP program to transfer the resulting compressed file.

17.2 Maintaining the Oracle BI Presentation Catalog

This section contains the following topics on maintaining a catalog:

- [Section 17.2.1, "Manually Changing Additional Configuration Settings for the Catalog"](#)
- [Section 17.2.2, "Manually Creating a New Oracle BI Presentation Catalog"](#)
- [Section 17.2.3, "Deploying Catalogs and Objects to Production"](#)
- [Section 17.2.4, "Validating the Catalog"](#)

17.2.1 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 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 Catalog 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: Keep the following points in mind:

- The HashUserHomeDirectories element must be set immediately after installing Oracle BI EE to be effective.
 - Include only one Catalog element in the instanceconfig.xml file or unexpected results might occur. Unless expressly noted, include most nodes in an XML document only once.
-
-

3. Include the element and its ancestor element as appropriate, as shown in the following example:

```
<ServerInstance>
<ps:Catalog xmlns:ps="oracle.bi.presentation.services/config/v1.1">
```

```
<ps:HashUserHomeDirectories>2</ps:HashUserHomeDirectories>
</ps:Catalog>
</ServerInstance>
```

Caution: Include only one Catalog element in the instanceconfig.xml file or unexpected results might occur. Unless expressly noted, include most nodes in an XML document only once.

4. Save your changes and close the file.
5. Restart Presentation Services.

17.2.2 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 10.2, "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.

When manually creating a new catalog in a clustered environment, you must restart each instance of Presentation Services separately using the Availability tab of the Capacity Management page in Fusion Middleware Control. Do not click **Restart** on the Business Intelligence Overview page. Wait a short time after restarting the first Presentation Services instance to give the system enough time to create the new catalog, then verify that the catalog was successfully created in the Presentation Services log file before restarting the other Presentation Services instances in the cluster.

17.2.3 Deploying Catalogs and Objects to Production

You can deploy catalogs and simple objects (for example, a dashboard with privileges) to a production environment from a test environment, as described in the following sections:

- [Section 17.2.3.1, "Deploying Catalogs to Production"](#)
- [Section 17.2.3.2, "Deploying Objects to Production"](#)

17.2.3.1 Deploying Catalogs to Production

Use the 7-Zip utility to archive an entire catalog in a test environment, then use the utility to unarchive the file in the production environment. Do not use Catalog Manager for archiving and unarchiving entire catalogs.

You might deploy a catalog in a cluster configuration in which different security environments are used. Some servers in the cluster might share the same LDAP server,

while others might not. If you deploy the catalog to a server that is communicating with a different LDAP server, then you must regenerate the user GUIDs for the catalog. See "Regenerating User GUIDs" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for information.

Optionally, if you are deploying the catalog from test to a new production environment, then use Fusion Middleware Control to specify the location of the new catalog in the production environment. For information, see [Section 10.2, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location."](#)

17.2.3.2 Deploying Objects to Production

You can deploy simple objects (for example, a dashboard with privileges) to a production environment from a test environment.

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 simple 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 17.10, "Archiving and Unarchiving Using Catalog Manager."](#)
 - b. Copy the archived file to the production computer.
 - c. On the production computer, unarchive the object.
For information about how to unarchive an object, see [Section 17.10, "Archiving and Unarchiving 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, then use Fusion Middleware Control to specify the location of the new catalog in the production environment.

For information, see [Section 10.2, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location."](#)

17.2.4 Validating the Catalog

Over time, inconsistencies can develop in the catalog as links are broken, users are deleted, or NFS file system issues are encountered. These inconsistencies can eventually lead to incorrect behavior, such as the inability to edit an agent's recipient list. You can periodically take the production system offline and validate the catalog, to be informed of and to take corrective action on inconsistencies.

17.2.4.1 Performing a Basic Validation of the Catalog

Use the following procedure to perform a basic validation of the catalog.

To validate the catalog:

1. Stop Presentation Services.

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

2. In 11g only, regenerate the user GUIDs for the catalog. See "Regenerating User GUIDs" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for information

Caution: Ensure that you regenerate the user GUIDs *before* validating the catalog. Failure to do so can result in the removal of all accounts, permissions, and privileges from the catalog.

3. Back up the catalog by using the 7-Zip utility to create a compressed file for it.
4. Create a backup copy of the instanceconfig.xml file.
For the location of the file, see [Section 3.6, "Where Are Configuration Files Located?"](#)
5. Edit the instanceconfig.xml file so that it contains the appropriate elements for performing the validation.
For information on these elements, see [Section 17.2.4.2, "Specifying the Elements for Validating the Catalog."](#)
6. Start Presentation Services to run the validation according to the values that you specified in the instanceconfig.xml file.
7. Stop Presentation Services.
8. Create a backup copy of the instanceconfig.xml file in which you added the validation elements, renaming the file similar to instanceconfig_validate.xml. In this way, you have a version of the file to use as a starting point for subsequent validations.
9. Restore the backup version of the instanceconfig.xml that you created earlier to use as the current version.
10. Start Presentation Services.

17.2.4.2 Specifying the Elements for Validating the Catalog

As part of the process of validating the catalog, you include elements in the `instanceconfig.xml` file that run the validation when you restart Presentation Services. The following procedure describes how to edit the `instanceconfig.xml` file to include these elements.

Before you begin this procedure, ensure that you are familiar with the information in [Section 3.4, "Using a Text Editor to Update Configuration Settings."](#)

To specify the element for validating the catalog:

1. Open the `instanceconfig.xml` file for editing as described in [Section 3.6, "Where Are Configuration Files Located?"](#)
2. Locate the Catalog section in which you must add the elements that are described in [Table 17-1](#).
3. Include the elements and their ancestor element as appropriate, as shown in the following example. In this example, the validation runs when Presentation Services starts. Inconsistent accounts (such as those for deleted users), links, and objects are removed. Inconsistent users' home directory names are logged but directories are not removed.

```
<ServerInstance>
<ps:Catalog xmlns:ps="oracle.bi.presentation.services/config/v1.1">
  <ps:Validate>OnStartupAndExit</ps:Validate>
  <ps:ValidateAccounts>Clean</ps:ValidateAccounts>
  <ps:ValidateHomes>Report</ps:ValidateHomes>
  <ps:ValidateItems>Clean</ps:ValidateItems>
  <ps:ValidateLinks>Clean</ps:ValidateLinks>
</ps:Catalog>
</ServerInstance>
```

Caution: Include only one Catalog element in the `instanceconfig.xml` file or unexpected results might occur. Unless expressly noted, include most nodes in an XML document only once.

4. Save your changes and close the file.

Table 17-1 Elements for Validating the Catalog

Element	Description	Default Value
Validate	<p>Performs the validation of the catalog according to the values of the other Validate-related elements in this section. Values are described in the following list:</p> <ul style="list-style-type: none"> ■ None — Performs no validation. ■ OnStartup — Performs the validation when Presentation Services starts. ■ OnStartupAndExit — Performs the validation when Presentation Services both starts and stops. This value allows re-running of the validation, performing as many cycles of Report and Clean until the catalog is as clean as appropriate. <p>If this value is not None, then all privileges and each object's ACLs in the entire catalog are cleaned of terminated accounts, regardless of the settings of the other Validate-related elements.</p>	None
ValidateAccounts	Verifies that all information about users, roles, and groups in the catalog is consistent. Values are described in the list after this table.	None
ValidateHomes	Verifies that all information about home directories in the catalog is consistent. Values are described in the list after this table.	None
ValidateItems	Verifies that all information about objects in the catalog is consistent. Values are described in the list after this table.	None
ValidateLinks	Verifies that all information about links in the catalog is consistent. Values are described in the list after this table.	None

The elements have the values that are described in the following list:

- None — Specifies that no validation is performed.
- Report — Specifies that details about each inconsistent object are written to the sawlog.log file.

For information, see [Section 8.3.2, "What Are Diagnostic Log Configuration Files and Where Are They Located?"](#)

- Clean — Specifies that details about each inconsistent object are written to the sawlog.log file and that each object is removed from the catalog.

17.3 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.

17.3.1 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 15.2.2, "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 "Managing Objects in the Oracle BI Presentation Catalog" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

17.3.2 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 catalog contents into e-mail, as this is not supported.

17.3.3 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.
- You can use Catalog Manager in languages other than English. For information, see [Section 15.3, "Setting the Current Locale in Catalog Manager."](#)

17.4 Starting Catalog Manager and Opening Catalogs

This section describes the following topics:

- [Section 17.4.1, "Starting Catalog Manager"](#)
- [Section 17.4.2, "Understanding the Two Catalog Modes"](#)
- [Section 17.4.3, "Operations Available in Online Mode and Offline Mode"](#)
- [Section 17.4.4, "Opening an Oracle BI Presentation Catalog"](#)

17.4.1 Starting Catalog Manager

Use the following procedure to start Catalog Manager.

To start Catalog Manager:

1. Using the command line, change to the following directory:

```
ORACLE_
```

```
INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplicati  
on_obipsn\catalogmanager
```

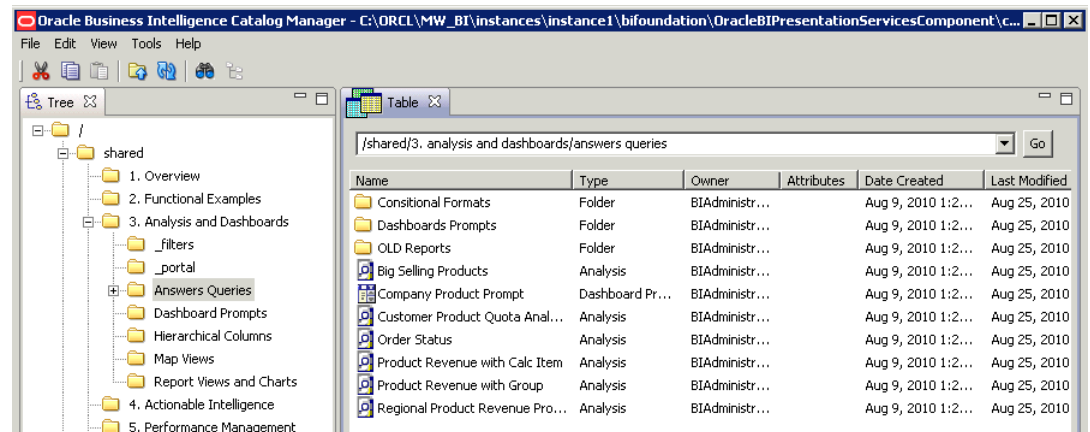
then run the appropriate script:

```
runcat.cmd (on Windows)
```

```
runcat.sh (on UNIX)
```

Figure 17–2 shows sample objects in the Catalog Manager for Windows.

Figure 17–2 Sample Objects in Catalog Manager



17.4.2 Understanding the Two Catalog Modes

You can open a catalog in one of two modes — online or offline. Both modes can operate against an actual production catalog, with no need for any downtime.

17.4.2.1 Online Mode

In online mode, you connect to a catalog on a running Web server. In this mode your permissions are applied, you can select a locale, and you can see the effects of any localization on the catalog. 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.

17.4.2.2 Offline Mode

In offline mode, you connect 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.

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.

Use offline mode when you want to make catalog-wide changes, such as globally renaming objects or moving multiple objects for reorganization. In a clustered environment, use the following procedure to make systemwide changes to the catalog.

To make systemwide changes to the catalog in a clustered environment:

1. Place Presentation Services on each computer in the cluster in Maintenance Mode. For information on Maintenance Mode, see "Administration page" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.
2. Back up the catalog by using the 7-Zip utility to create a compressed file for it.
3. In Catalog Manager, open the catalog in offline mode, as described in [Section 17.4.4, "Opening an Oracle BI Presentation Catalog."](#)
4. Make the systemwide change.
5. Back up the catalog again.
6. Take Presentation Services on each computer in the cluster out of Maintenance Mode.

17.4.3 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.

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.
- Renaming objects with reference updates. (This feature is known as Smart Rename and is available in both modes. In offline mode, you can rename all objects. In online mode, you might be unable to rename certain objects, depending on your permissions.)
- 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.
- Searching for and replacing catalog text. (This feature is available in both modes. In offline mode, you can replace all objects. In online mode, you might be unable to replace certain objects, depending on your permissions.)
- Creating reports to display Catalog Manager data.
- Setting browser preference.
- Previewing objects (available in online mode only).
- Exporting captions for localization purposes.

17.4.4 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, as 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\coreappl
ication_obipsn\catalog\default
 - **URL** — If you are opening the catalog in online mode, then enter the URL to Oracle BI Presentation Services, for example:
https://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).
 - **Locale** — Select the locale to use for user interface elements in Catalog Manager and for objects in the catalog, when opening the catalog in online mode.
 - **Read-Only** — Select this field to open the catalog in read-only mode (disabled in offline mode).

3. Click **OK**.

When specifying the URL for the catalog in online mode, ensure that you specify `https` rather than `http`, for increased security. If you specify `http`, then you see a message box after closing the Open Catalog dialog that prompts you to verify the opening of the catalog using an unsecured connection. To use `https` when opening catalogs, you must configure Oracle BI EE for Secure Socket Layer communication, as described in "SSL Configuration in Oracle Business Intelligence" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

17.5 Using the Catalog Manager Workspace

This section provides the following topics on the workspace for Catalog Manager:

- [Section 17.5.1, "What Does the Catalog Manager Workspace Do?"](#)
- [Section 17.5.2, "What Does the Catalog Manager Workspace Look Like?"](#)
- [Section 17.5.3, "Managing the View of the Catalog Manager Workspace"](#)

17.5.1 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 configured by you as the administrator, such as privileges. Avoid modifying any files in this folder. Presentation Services uses these files internally and modifying them might cause unexpected results.
- The users folder — Contains content that catalog users with the appropriate permissions have saved to their personal folders, such as individual analyses.

17.5.2 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 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 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.)

17.5.3 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, select **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.)

17.6 Working with Objects in Catalog Manager

This section provides the following information about working with objects:

- [Section 17.6.1, "Searching for Catalog Objects Using Catalog Manager"](#)
- [Section 17.6.2, "Copying and Pasting Objects"](#)
- [Section 17.6.3, "Renaming Catalog Objects"](#)
- [Section 17.6.4, "Working with the Properties of Catalog Objects"](#)
- [Section 17.6.5, "Setting Permissions of Catalog Objects"](#)
- [Section 17.6.6, "Previewing Objects from Catalog Manager"](#)
- [Section 17.6.7, "Working with Tenants"](#)

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 "Managing Objects in the Oracle BI Presentation Catalog" in *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.

17.6.1 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.

17.6.2 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.

17.6.2.1 Tips for Copying and Pasting

Use the following tips as you copy and paste objects:

- You can copy and paste objects using the following methods:
 - Menu options, as described in [Section 17.6.2.1.1, "Copying and Pasting Objects Between Catalogs Using Menus."](#)
 - Drag and drop, 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.
 - Archive and unarchive, as described in [Section 17.10, "Archiving and Unarchiving Using Catalog Manager."](#) When you archive, you create a file that you can save for later use. The unarchiving process automatically overwrites any files without providing the opportunity to specify that certain files not be overwritten.
- 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 17.6.2.2, "Advanced Options for Pasting Objects."](#)

17.6.2.1.1 Copying and Pasting Objects Between Catalogs Using Menus 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**.

17.6.2.2 Advanced Options for Pasting Objects

You can set advanced options in the Preferences dialog for pasting objects that you have copied, as described in the following sections:

- [Section 17.6.2.2.1, "Paste Overwrite"](#)
- [Section 17.6.2.2.2, "Paste ACL"](#)

Important: You must set the advanced options in the Preferences dialog before you begin the copy and paste operation, for them to take effect.

17.6.2.2.1 Paste Overwrite The Preferences dialog contains the following options in the Paste Overwrite area:

- **Force** — Pastes all files, overwriting even those that have the read-only attribute set.
- **All** — Pastes all possible files, overwriting only those that do not have the read-only attribute set. (Default)
- **Old** — Pastes all possible files, but does not overwrite any existing files unless they are older than the source.
- **None** — Pastes all possible files, but does not overwrite 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.

17.6.2.2.2 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, depending on the mode and type of owner, as described in the following list:
 - **Online mode** — In online mode, Catalog Manager is communicating with the back-end security server. Catalog Manager knows about the users and application roles from that server and can usually paste a copied object with the appropriate user name or role. While pasting objects, keep in mind that you might lack appropriate permissions to create accounts for certain objects.
 - **Offline mode** — In offline mode, Catalog Manager has no connection with the back-end security server, so it is unaware of users and application roles that are stored there, unless their names are available in the cache for the catalog. If the name of the user or role for a copied object is not available in the cache, then Catalog Manager cannot paste the copied object with that name or role. Instead, the pasted object inherits its owner from its new parent folder, which is similar to the **Inherit** option.

However, because Catalog groups are stored in the catalog rather than in the back-end security server, then Catalog Manager can associate Catalog groups with pasted objects as appropriate.

This feature is used in applications whose administrators create accounts in a staging area before moving the users to the production environment.

- **Create Only Groups** — Same as **Create**, but applies only to Catalog groups, not to user accounts. Works in a development to production environment similarly to **Preserve Only Groups**.

If you have the appropriate permissions, then you can select a newly pasted object and set ownership recursively to the appropriate user.

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 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** in online mode, 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 owner is inherited from the parent folder. (The Create option is deprecated in Release 11g as it applies only to Catalog groups.)

17.6.3 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). This feature is also known as Smart Rename. You can open the catalog in either offline or online mode. In offline mode, you can rename all objects. In online mode, you might be unable to rename certain objects, depending on your permissions.

Caution: Keep the following points in mind when renaming objects:

- You cannot rename a user account in the catalog. If you rename a user's home directory, then you do not rename that user and you might see unexpected results.
 - The catalog contains several reserved names of objects. Rename only your own objects, not those that Presentation Services creates internally. For example, do not rename the `_portal` directory in your home directory, because then you cannot see "My Dashboard".
-
-

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 corner of the window shows the progress of the reference updates.

17.6.4 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.

The Owner list includes the name that you used to log in to Catalog Manager. You can use this list to select yourself as the owner of the object.

See "Assigning Ownership of Objects" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for more information on taking ownership of objects.

- 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 Services.

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**. You can also select **Apply Recursively** to take ownership of an object and all its sub-objects.

17.6.5 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, Catalog Groups and Application Roles (Explicit Permissions)** — Shows the users, groups, and application roles that have explicit permissions granted to this object.

- **Additional Users and Application Roles** — 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 "Managing Security for Dashboards and Analyses" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

4. If the user, group, or application role whose permissions you want to set is in the **Additional Users and Application Roles** list, then move it into the **Users, Catalog Groups and Application Roles (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 Application Roles** list, use the **List** button and the adjacent field, as follows:
 - Enter filter criteria in the field next to the **List** button (case insensitive) to search by name.
To enter partial filter criteria, use the asterisk (*) symbol. For example, enter bi* to display users or groups beginning with bi, BI, bI, and Bi.
 - Select a value from the list, to restrict what accounts to search for.
Available values are: All, User, Catalog Group, or Application Role.
6. Select the user or group in the **Users, Catalog Groups and Application Roles (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 "Permission Definitions" in *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 Application Roles** 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 *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for more information about permissions and groups and users.

17.6.6 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**.

17.6.7 Working with Tenants

Catalog Manager provides options for provisioning and unprovisioning tenants. This section includes the following topics:

- [Section 17.6.7.1, "What Happens When Tenants are Provisioned?"](#)
- [Section 17.6.7.2, "What Options are Available for Provisioning?"](#)
- [Section 17.6.7.3, "How is Multi-Tenancy Enabled?"](#)

17.6.7.1 What Happens When Tenants are Provisioned?

For example, you can provision certain companies if they sign a contract with your organization. Suppose that TenantA is provisioned. A copy of the /shared directory is included in the /tenants/TenantA/shared directory, and the home directories for TenantA's users are included in the /tenants/TenantA/users directory. These inclusions are completely transparent to the TenantA users. They continue to see their objects stored as if they are in the /shared and /users directories.

17.6.7.2 What Options are Available for Provisioning?

You can use either menu options or the command line to provision tenants. From the Tools menu in Catalog Manager, select **Multi-Tenancy**, then either **Provision Tenant** or **Unprovision Tenant**.

You can also use the provisionTenant command. The syntax is as follows:

```
runcat.cmd/runcat.sh -cmd provisionTenant -offline
catalog-path-name -tenants tenant-ID
```

The following is a sample command on Windows:

```
runcat.cmd -cmd provisionTenant -offline
C:\Oracle\Middleware\instances\instance1\bifoundation\OracleBIPr
esentationServicesComponent\coreapplication_
obips1\catalog\SampleAppLite -tenants
49431843829925000:49431843829924000
```

17.6.7.3 How is Multi-Tenancy Enabled?

To enable the options for multi-tenancy, you must include the appropriate element in the instanceconfig.xml file, as described in the following procedure.

To enable the use of multi-tenancy:

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 EnableMultiTenancy element.

The EnableMultiTenancy element is set to false by default. If you do not set it to true, then the options for multi-tenancy are not available.

3. Include the element and its ancestor elements as appropriate, as shown in the following example:

```
<ps:Catalog>
  <ps:EnableMultiTenancy>true</ps:EnableMultiTenancy>
</ps:Catalog>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

17.7 Viewing and Editing Catalog Objects in XML

Catalog Manager provides the ability to view and to edit the XML description of catalog objects such as analyses, dashboards, filters, and so on. While viewing the XML code is acceptable, editing the code is not recommended.

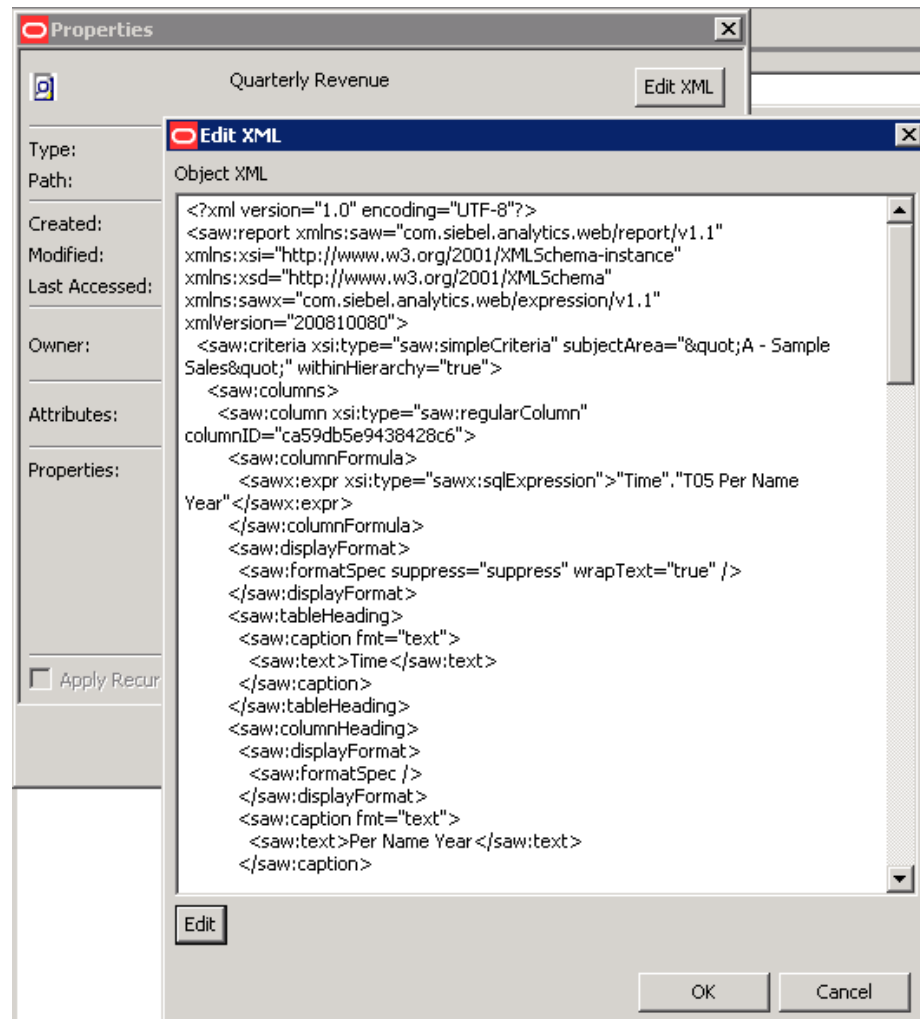
Caution: If you edit the XML code, then you change the representation of the object in the catalog. Editing the XML code for catalog objects in any directory is not recommended and can produce unexpected results.

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.

[Figure 17-3](#) shows sample XML code in Catalog Manager for an object.

Figure 17-3 Sample XML Code for an Object



To edit the XML description of an object, which is not recommended:

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.

17.8 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. In offline mode, you can replace all objects. In online mode, you might be unable to replace certain objects, depending on your permissions.

Specifically, you can search for and replace:

- A simple text string using a dialog, as described in [Section 17.8.1, "Searching for and Replacing a Simple Catalog Text String."](#)

For example, suppose that an object contains the string "My Misspeled Wirds." You can use Catalog Manager to search and replace that string with the proper text of "My Misspelled Words."

- Multiple or complex text strings all at the same time using an XML file, as described in [Section 17.8.3, "Searching for and Replacing Multiple Catalog Text Strings."](#)

For example, suppose that the administrator renames a subject area, a table, or column in the repository file. The table "Sales" might be renamed "MySales." You can use Catalog Manager to search and replace all uses of that object throughout the catalog.

17.8.1 Searching for and Replacing a Simple Catalog Text String

Use the following procedure to search for a simple text string in the catalog and replace it with other text.

To search for and replace a simple 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, deselect the **Case Sensitive** box.
6. Click **OK**.

17.8.2 About Searching for and Replacing Multiple Catalog Text Strings

You can perform more powerful search and replace operations on multiple catalog text strings all at the same time by importing a XML file that identifies each text string to search for and replace.

17.8.2.1 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 the text to replace. The valid value is:
 - **textReplace** — Replaces all the text that matches in an XML file, such as a column name.
- **oldValue** — Specifies the text string to search for.

When you specify this attribute for the textReplace command for the search and replace XML file, you must use the full Java regex syntax, which is not like a normal string. To replace a string, you must do the following:

1. Escape any special Java regex characters (such as brackets, parentheses, dollar signs, and carets).
2. Escape any special "normal" string characters (such as backslashes and quotes).
3. Because you are working in an XML file, escape any special HTML characters (such as quotes and ampersands).

The full Java regex syntax is described in the following document:

<http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html>

Table 17-2 provides sample strings for use with the regex syntax in search criteria.

Table 17-2 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".
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 asterisk (*) for example, "a*".
?	Use a question mark (?) with a character and an asterisk (*) to return zero (0) or more occurrences of a character. For example ?a* returns zero or more occurrences of the character "a".

- 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.

17.8.2.2 Example XML File for Searching for and Replacing Text Strings

The following is a partial example of an XML file for searching for and replacing a text string:

```
<?xml version="1.0" encoding="utf-8"?>
<actions>
<action command="textReplace" oldValue="boots" newValue="HoleyShoes"
ignoreCase="true"/>
</actions>
```

17.8.3 Searching for and Replacing Multiple Catalog Text Strings

Use the following procedure to search for and replace multiple catalog text strings all at the same time.

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 17.8.2, "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**.

17.9 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 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. Repeat Steps 4 through 7 until you the report contains the appropriate columns.
9. 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).
10. Click **OK** to overwrite an existing file or click **Cancel** and enter a new name to save to a new file.
11. Select **Excel Format** to specify to create a file with a .tab extension that can be imported into Microsoft Excel 2003 or later.
12. Click **OK**.

17.10 Archiving and Unarchiving Using Catalog Manager

Catalog Manager provides the ability to archive and unarchive either an individual catalog folder or an entire catalog. See the following list for important information on this functionality:

- When you archive 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.
- When you unarchive 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 without warning.
- Archiving and unarchiving an entire catalog using Catalog Manager or the Catalog page in Presentation Services is not recommended. Instead, use either the 7-Zip utility or .tar.gz files on UNIX systems for archiving and unarchiving an entire catalog.
- Do not use the following utilities when archiving and unarchiving an entire catalog:
 - WinZip — The WinZip utility does not always handle extended UNIX file permissions properly. Using the WinZip utility to move catalogs in heterogeneous environments might lead to corrupted catalog files.
 - FTP — Some third-party FTP programs rename internal files, which might lead to corrupted catalog files. Before moving a catalog using an FTP program, use the 7-Zip utility to compress the catalog into one file.

You can use the Catalog page in Presentation Services to archive and unarchive objects. For information, see "Archiving Objects" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

17.10.1 Archiving a Folder Using Catalog Manager

Use the following procedure to archive a catalog folder.

To archive an individual catalog folder in the catalog to a file that you specify:

1. In Catalog Manager, open the catalog in offline mode.
2. 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 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 and the **Old** option in the Paste Overwrite area of the Preferences dialog is ignored. Upon unarchiving, the system applies a timestamp that indicates the time at which the object or folder is unarchived.

See [Section 17.6.2.2, "Advanced Options for Pasting Objects"](#) for more information.

- 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 and the options in the Paste ACL area of the Preferences dialog are ignored. Upon unarchiving, the system assigns the parent folder's permissions to all of the objects and folders.

5. Click **OK**.

17.10.2 Unarchiving a Folder Using Catalog Manager

Unarchiving is similar to pasting and therefore requires that you understand the issues that relate to permissions and ACLS as described in [Section 17.6.2.2, "Advanced Options for Pasting Objects."](#) Use the following procedure to unarchive a catalog folder.

To unarchive a catalog folder:

1. In Catalog Manager, open the catalog in offline mode.
2. To unarchive a catalog folder, 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 folder to unarchive. Click **Browse** to display a dialog for selecting the path name.
5. Click **OK**.

17.11 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 17.11.1, "What is a Full-Text Catalog Search?"](#)
- [Section 17.11.2, "Deployment of Oracle Full-Text Catalog Search"](#)
- [Section 17.11.3, "How is Security Handled in the Full-Text Catalog Search?"](#)
- [Section 17.11.4, "Process for Configuring the Full-Text Catalog Search"](#)

For information about content designers and users using the full-text catalog search, see "Searching Using Full-Text Catalog Search" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

17.11.1 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. This full-text search provides more advanced options than the basic catalog search that Presentation Services provides, as described in the following list:

- **Basic Catalog Search** — Users can search for an object by its exact name, which is similar to using a Find dialog in many products.
- **Full-text Catalog Search** — Users can search for objects by various attributes, such as name, description, author, and the names and values of columns of data that the object references. Users obtain search results that correspond to the objects that they have access to open.

For more information on the types of searches, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

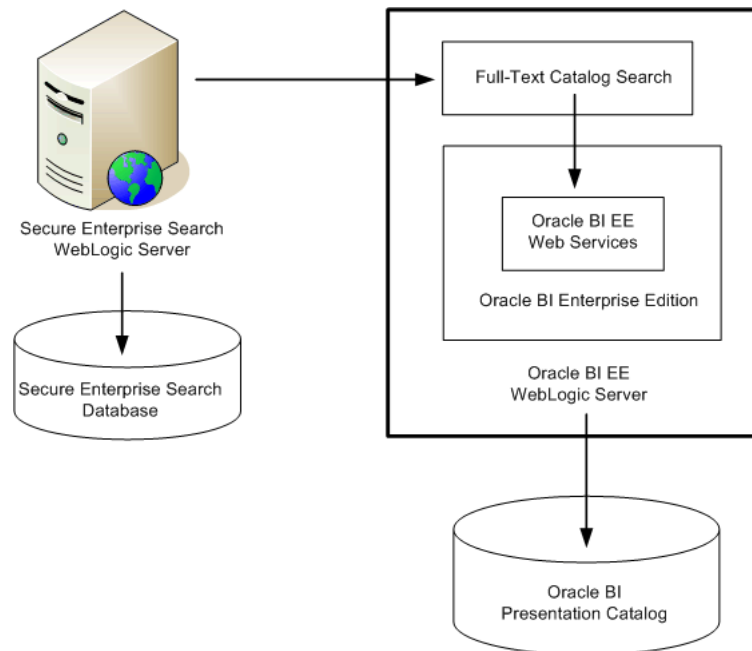
As the administrator, you must configure the use of the full-text catalog search. For this release, the full-text catalog search uses a *crawler* for Secure Enterprise Search. The full-text catalog search can index catalogs that contain translated text. Secure Enterprise Search limits each of its registered 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.

17.11.2 Deployment of Oracle Full-Text Catalog Search

Figure 17-4 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 17-4 Deployment of the Full-Text Catalog Search



17.11.3 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.

17.11.4 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"](#)

17.11.4.1 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 about assigning privileges, see "Managing Presentation Services Privileges" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

17.11.4.2 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 or later. 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 about SES, display the following page from Oracle Technology Network:

<http://www.oracle.com/us/products/database/secure-enterprise-search/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.

17.11.4.3 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**.

17.11.4.4 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 EE, in the following format:
`http://host:port/bisearch/urlbuilder` (for Secure Enterprise Search 11.1.2)
`http://host:port/bisearch/urlbuilder/` (for Secure Enterprise Search 11.2.1 or later)
 where *host* is the computer on which Oracle BI Enterprise Edition is installed and *port* is its port number.
6. Click one of the following:
 - Click **Create** to automatically launch the schedule.
 - Click **Create and Customize** to perform any customizations, such as including additional documents to be processed (for example, RTF documents), as described in the next step.
7. If you clicked **Create and Customize**, then on the Documents tab, add the **RTF Document** document type and any other appropriate types to the "Processed" document types list.

8. Click **Apply**.

17.11.4.5 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 [Section 17.11.4.7, "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 "Accessing Properties" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

17.11.4.6 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.

17.11.4.7 Step 7: Enabling Incremental Crawls for Full-Text Catalog Search

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 about 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 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`

17.11.4.8 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 17-3](#) 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 EE.

Table 17-3 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
BIServerSSOURL	Specifies the SSO URL for the Oracle HTTP Server instance that is configured to handle URL requests for Oracle BI EE.	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 <i>not</i> 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.	"/"

17.11.4.9 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.

17.11.4.9.1 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.

17.11.4.9.2 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 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 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/search/query/search</ps:SESSearchURL>
</Server>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

17.12 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 perform this sharing, 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 about these SOAP calls, see *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition*.

17.12.1 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 about this element, see [Section 17.12.6, "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.

17.12.1.1 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 17-4](#).

Table 17-4 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

17.12.2 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 17.12.3, "Copying Oracle BI Presentation Catalog Files"](#)
- [Section 17.12.4, "Using the Oracle BI Presentation Services Replication Agent"](#)
- [Section 17.12.5, "Creating the config.xml File for Replication"](#)
- [Section 17.12.6, "Editing the instanceconfig.xml File for Replication"](#)

17.12.3 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.

17.12.4 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 17.12.5, "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 17.12.4.1, "mark"](#)
- [Section 17.12.4.2, "remotecopy"](#)
- [Section 17.12.4.3, "run"](#)

17.12.4.1 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 /
```

17.12.4.2 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
```

17.12.4.3 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
```

17.12.5 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 17.12.5.1, "Structure of the Configuration File"](#)
- [Section 17.12.5.2, "Elements of the Configuration File"](#)
- [Section 17.12.5.3, "Example config.xml File"](#)

For information about manual editing, see [Section 3.4, "Using a Text Editor to Update Configuration Settings."](#)

17.12.5.1 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>
```

17.12.5.2 Elements of the Configuration File

[Table 17-5](#) 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 17-5 Replication Agent Configuration Elements in config.xml

Element	Description
Config	Serves as the root element for the configuration file.

Table 17-5 (Cont.) Replication Agent Configuration Elements in config.xml

Element	Description
Config/General	Contains general settings that are applicable to all instances and replication tasks.
General/ExportDirectory	<p>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.</p> <p>In addition, the user account under which an instance of Presentation Services is running must have read/write permissions to this directory.</p>
General/LogExpiresHours	<p>Specifies the time in hours (as a double value) that replication logs expire on all instances of Presentation Services.</p> <p>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.</p> <p>For information about the ReadLogRecordsSinceHoursAgo element, see Section 17.12.6, "Editing the instanceconfig.xml File for Replication."</p>
General/RetryAttempts	<p>Specifies the number of times an attempt to change an object is to be retried.</p> <p>(An attempt to change an object might be retried, for example, if an object was locked.)</p>
Config/Server	<p>Defines connection information for each computer for Presentation Services. Its attributes are:</p> <ul style="list-style-type: none"> ▪ 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 17-5 (Cont.) Replication Agent Configuration Elements in config.xml

Element	Description
Config/Server/Property	<p>Defines connection parameters for the Presentation Services (SAWBridge) server and the transport protocol. Its attributes are:</p> <p>Defines server name and value:</p> <p>Note: The settings that you use here are determined by whether the value for the transport protocol is set to use HTTP (N) or socket based TCP (Y).</p> <ul style="list-style-type: none"> ▪ 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). <p>Defines the transport protocol name and value:</p> <ul style="list-style-type: none"> ▪ name - The transport protocol name, for example: oracle.bi.presentation.usesoaptcptransport ▪ value - The transport protocol for connecting to the Presentation Services can be either of the following: N - Use HTTP protocol (Default value). Y - Use socket based TCP protocol.
Config/Folderset	<p>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".</p>
Folderset/Folder	<p>Adds a folder to the folderset. Specifies the full path to the catalog folder.</p>
Config/ReplicationTask	<p>Defines the replication tasks. Its attributes are:</p> <ul style="list-style-type: none"> ▪ 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)

17.12.5.3 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>
```

17.12.6 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 about 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 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 17–6](#).
3. Include the elements and their ancestor elements as appropriate, as shown in [Example 17–1](#).

Note: Do not insert multiple Catalog elements into a single instanceconfig.xml file. Search for an existing Catalog element and if it exists, use that element for holding the elements that are shown in this example.

4. Save your changes and close the file.
5. Restart Presentation Services.

Table 17–6 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 which of the IDs that were specified in the IDsInCluster element to use as the suffix for log files that are generated by this instance of Presentation Services. Log files are named as follows: <ul style="list-style-type: none"> ■ For change log files: sawchange_InstanceID_SequenceNum.log ■ For playback log files: sawplayback_InstanceID_SequenceNum.log See Table 17–7 for example file names. This element is required in a clustered environment.	NA

Example 17–1 instanceconfig.xml Files on Multiple Servers for Replication

Suppose that you have two servers for which you want to enable replication. You set the instanceconfig.xml file to the following values on Server1:

```
<ServerInstance>
<ps:Catalog xmlns:ps="oracle.bi.presentation.services/config/v1.1">
  <Replication>
    <Enabled>true</Enabled>
```

```

<ReadLogRecordsSinceHoursAgo>120</ReadLogRecordsSinceHoursAgo>
<RecordsInFileLimit>4000</RecordsInFileLimit>
<Cluster>
  <IDsinCluster>ID_FOR_S1, ID_FOR_S2</IDsinCluster>
  <MyInstanceID>ID_FOR_S1</MyInstanceID>
</Cluster>
</Replication>
</ps:Catalog>
</ServerInstance>

```

On Server2, you set the instanceconfig.xml file to these same values, with one exception. You set the MyInstanceID element to a value of ID_FOR_S2 to indicate that log files that are generated by the instance of Presentation Services on this server use a suffix of ID_FOR_S2.

```
<MyInstanceID>ID_FOR_S2</MyInstanceID>
```

Table 17-7 shows sample replication log file names that are generated by Server1 and Server2. Each node in the cluster must know the names of the other server log files to read replication information that is generated by the other servers.

Table 17-7 Replication Log File Names by Server

Log File Type	Server1	Server2
Change log	sawchange_ID_FOR_S1_1.log	sawchange_ID_FOR_S2_1.log
	sawchange_ID_FOR_S1_2.log	sawchange_ID_FOR_S2_2.log
	sawchange_ID_FOR_S1_3.log	sawchange_ID_FOR_S2_3.log
	sawchange_ID_FOR_S1_4.log	sawchange_ID_FOR_S2_4.log
Playback log	sawplayback_ID_FOR_S1_1.log	sawplayback_ID_FOR_S2_1.log
	sawplayback_ID_FOR_S1_2.log	sawplayback_ID_FOR_S2_2.log
	sawplayback_ID_FOR_S1_3.log	sawplayback_ID_FOR_S2_3.log
	sawplayback_ID_FOR_S1_4.log	sawplayback_ID_FOR_S2_4.log

17.12.7 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 17.12.6, "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. Reenable 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 includes the following chapters:

- [Chapter 18, "Configuring and Managing Analyses and Dashboards"](#)
- [Chapter 19, "Configuring and Managing Agents"](#)
- [Chapter 20, "Configuring Advanced Options for Mapping and Spatial Information"](#)
- [Chapter 21, "Configuring Resource Availability and URL Generation"](#)

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 about 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 includes the following topics:

- [Section 18.1, "Managing Dashboards"](#)
- [Section 18.2, "Performing General Configuration Tasks for Analyses"](#)
- [Section 18.3, "Configuring for Displaying and Processing Data in Views"](#)
- [Section 18.4, "Manually Changing Presentation Settings"](#)
- [Section 18.5, "Blocking Analyses in Answers"](#)
- [Section 18.6, "Specifying View Defaults for Analyses and Dashboards"](#)
- [Section 18.7, "Configuring for Write Back in Analyses and Dashboards"](#)

18.1 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 about dashboards:

- Guidelines for creating a shared dashboard, within the broader context of the Oracle BI Presentation Catalog structure and security framework, are provided in "Controlling Access to Saved Customization Options in Dashboards" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
- Information about shared folder structures in the Oracle BI Presentation Catalog is provided in [Chapter 17, "Configuring and Managing the Oracle BI Presentation Catalog."](#)

- Information about permissions is provided in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
- Details for enabling users to act for others, which allows them to access the other users' dashboards, is provided in "Enabling Users to Act for Others" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

18.2 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 18.2.1, "Providing Access to Metadata Dictionary Information"](#)
- [Section 18.2.2, "Supporting Nested Folders, Navigation, and Drill-Down"](#)

18.2.1 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, see "Generating a Metadata Dictionary" in *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 about 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/demo1</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 Configuration Settings."](#)

3. Grant the Access to Metadata Dictionary privilege to the appropriate content designers. For information about privileges, see "Managing Presentation Services Privileges" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

For details on how content designers can view metadata dictionary information, see "Viewing Metadata Information from the Subject Areas Pane" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

18.2.2 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 about setting up subject areas. The following list includes 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.

- Content designers can create analyses that include columns from a primary subject area and from one or more related subject areas.

18.3 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 18.3.1, "Manually Configuring for Data in Views"](#)
- [Section 18.3.2, "Manually Configuring for Graphs and Gauges"](#)
- [Section 18.3.3, "Manually Changing Alternating Bar Color"](#)

18.3.1 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 18.3.1.1, "Manually Configuring Cube Settings for Pivot Tables and Graphs"](#)
- [Section 18.3.1.2, "Manually Configuring Settings for Data in Views"](#)

18.3.1.1 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 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.

18.3.1.2 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 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 18–1](#).
3. Include the elements and their ancestor elements as appropriate, as shown in the following example.

```
<ServerInstance>
  <Views>
    <Table>
      <MaxCells>10000</MaxCells>
      <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 18–1](#) 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 18–1 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 views in analyses and dashboards. This number should not exceed the number that is specified for the MaxVisibleRows element.	25	Pivot Table, Table
MaxCells	Specifies the maximum number of cells to be displayed in a view. This number should not exceed the product of MaxVisibleColumns times MaxVisibleRows, which is what the system attempts to render.	50000	Pivot Table, Table
MaxVisibleColumns	Specifies the maximum number of columns to be displayed in a view.	300	Graph, Pivot Table
MaxVisibleRows	Specifies the maximum number of rows to be displayed in a view. The value of DefaultRowsDisplayed should not exceed this value. For tables and pivot tables, specifies the following: <ul style="list-style-type: none"> The number of rows that is displayed on the tooltip for the Display Maximum Rows per Page paging control button. The uppermost value to specify for the Maximum Number of Rows to Download and the Maximum Number of Rows Per Page to Include options in Fusion Middleware Control. For information, see Section 7.3.3, "Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables." 	500	Graph, Pivot Table, Table
MaxVisiblePages	Specifies the maximum number of view prompts (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 18–2 .	25	Graph, Pivot Table, Table
JavaHostReadLimitInKB	Specifies the maximum amount of data that is sent to the browser for a single graph.	4096	Graph

18.3.2 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 18.3.1, "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 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 18–2](#).
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>
      <DefaultWebImageType>png</DefaultWebImageType>
      <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 18–2 Common Elements for Manually Configuring Graphs and Gauges

Element	Description	Default Value
EmbedFonts	See Section 18.3.2.1, "Configuring Fonts for Graphs" for details.	False
DefaultWebImageType	Specifies the default type for rendering an image when a format has not been specified in the URL or in the XML file for the view. Valid values are flash, png, and svg. (The svg value is not supported in this release, so flash is used if svg is specified.)	flash
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 about defining section sliders in graphs and gauges, see <i>Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition</i> .	5
SectionSliderLimit	Specifies the maximum number of values that can be displayed on a section slider bar.	10

Table 18–2 (Cont.) Common Elements for Manually Configuring Graphs and Gauges

Element	Description	Default Value
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

18.3.2.1 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

18.3.2.1.1 Setting the Embed Fonts Element By default, graphs rely on users to have the appropriate device fonts installed on their system to display multilingual 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.

18.3.2.1.2 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 about font configuration files, see your Java documentation.

18.3.3 Manually Changing Alternating Bar Color

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 under 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>
```

18.4 Manually Changing Presentation Settings

You can configure settings that change the display of dashboards and presentation settings, as described in the following sections:

- [Section 18.4.1, "Manually Changing Presentation Setting Defaults"](#)
- [Section 18.4.2, "Configuring Links to Dashboard Pages"](#)
- [Section 18.4.3, "Configuring an Alternate Toolbar for Oracle BI Publisher"](#)
- [Section 18.4.4, "Modifying the Table of Contents for PDF Versions of Briefing Books"](#)

18.4.1 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 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 18–3.](#)
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 18.4.2, "Configuring Links to Dashboard Pages"](#) for information about the Bookmarks, MaxAgeMinutes, EnableBookmarkURL, and EnablePromptedURL elements.
7. See [Section 18.4.3, "Configuring an Alternate Toolbar for Oracle BI Publisher"](#) for information about the ReportingToolBarMode element.

Table 18–3 Common 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

Table 18–3 (Cont.) Common Elements for Manually Changing Additional Presentation Setting Defaults

Element	Description	Default Value
MaxDropDownValues	<p>Specifies the maximum number of choices to display in the following locations:</p> <ul style="list-style-type: none"> ■ 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. <p>For information about prompts and searching, see <i>Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition</i>.</p>	256
MaxFollowLinks	<p>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.</p> <p>The default value for this element is 5; the minimum is 1; and the maximum is 10.</p> <p>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 18.4.4, "Modifying the Table of Contents for PDF Versions of Briefing Books."</p> <p>For information about working with briefing books, see <i>Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition</i>.</p>	5

18.4.2 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 under 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>
```

18.4.3 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 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.

18.4.4 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 "Working with Briefing Books" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for information about the table of contents.

The default template for the table of contents, `toc-template.rtf`, is located in the `ORACLE_INSTANCE\config\OracleBIPresentationServicesComponent\coreapplication_obisn` directory. You can modify the `toc-template.rtf` file to accommodate the needs of your organization.

18.5 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 18.5.1, "Storing JavaScript Files"](#)
- [Section 18.5.2, "Blocking Analyses Based on Criteria"](#)
- [Section 18.5.3, "Blocking Analyses Based on Formula"](#)

- [Section 18.5.4, "Validation Helper Functions"](#)

18.5.1 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_
obipsn\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" />
```

18.5.2 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`.

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_`
`obipsn\analyticsRes` directory. The following is sample code for the `myblocking.js` program.

```
// This is a blocking function. It ensures that users select what
// the designer wants them to.
function validateAnalysisCriteria(analysisXml)
{
  // Create the helper object
  var tValidator = new CriteriaValidator(analysisXml);
  // Validation Logic
  if (tValidator.getSubjectArea() != "Sample Sales")
    return "Please try Sample Sales?";
}
```

```

if (!tValidator.dependentColumnExists("Markets", "Region", "Markets", "District"))
{
    // If validation script notifies user, then return false
    alert("Region and District are well suited, do you think?");
    return false;
}
if (!tValidator.dependentColumnExists("Sales Measures", "", "Periods", "Year"))
return "You selected 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 are 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.

18.5.3 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 about the helper functions, see [Section 18.5.4, "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)
{
  // 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.

18.5.4 Validation Helper Functions

These functions are defined within a JavaScript file named `answers/queryblocking.js`. [Table 18–4](#) contains the list of helper functions and their descriptions.

Table 18–4 Validation Helper Functions

Validation Helper Function	Description
<code>CriteriaValidator.getSubjectArea()</code>	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.
<code>CriteriaValidator.tableExists(sTable)</code>	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.
<code>CriteriaValidator.columnExists(sTable, sColumn)</code>	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 18–4 (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.

18.6 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.

18.6.1 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.

18.6.2 Examples of Customizing Default Values for Analyses and Dashboards

The following sections provide examples of customizing default values:

- [Section 18.6.2.1, "Adding a Default Header or Footer to New Analyses"](#)
- [Section 18.6.2.2, "Preventing Auto-Previewing of Results"](#)
- [Section 18.6.2.3, "Setting Defaults for Analyses in the Compound Layout"](#)
- [Section 18.6.2.4, "Changing Dashboards Section Defaults"](#)
- [Section 18.6.2.5, "Specifying Dashboard Page Defaults Including Headers and Footers"](#)

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.

18.6.2.1 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>
```

18.6.2.2 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" />
</HTML>
</WebMessage>
  </WebMessageTable>
</WebMessageTables>
```

```

    <view signature="htmlview" autoPreview="false" />
</HTML>
</WebMessage>
  </WebMessageTable>
</WebMessageTables>

```

18.6.2.3 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>

```

18.6.2.4 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>

```

18.6.2.5 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>
```

18.7 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 about how you as the administrator can configure for write back:

- [Section 18.7.1, "Write-Back Limitations"](#)
- [Section 18.7.2, "Configuring for Write Back"](#)
- [Section 18.7.3, "About the Write-Back Template"](#)

18.7.1 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.

18.7.2 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 nonnullable.

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 18.7.3, "About the Write-Back Template."](#)
4. Add the LightWriteback element in the instanceconfig.xml file, as described in [Section 18.7.2.1, "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 "Managing Presentation Services Privileges" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
6. Configure a write-back analysis, as described in "Process for Write Back" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

18.7.2.1 Setting the LightWriteback Element

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 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.

18.7.3 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.

18.7.3.1 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.

18.7.3.2 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 quotation marks 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/coreapplicati
on_obipsn/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.

- 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">
```

18.7.3.3 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>
```

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 "Delivering Content" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

This chapter includes the following topics:

- [Section 19.1, "How Are Agents Used?"](#)
- [Section 19.2, "How Do Antivirus Software and Privileges Affect Agents?"](#)
- [Section 19.3, "Configuring Settings That Affect Agents"](#)
- [Section 19.4, "Managing Device Types for Agents"](#)
- [Section 19.5, "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 8.5.6, "Diagnosing Issues with Agents"](#) for information about diagnostics and log files for agents.

19.1 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, see "What is Oracle BI Scheduler?" in *Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition*

19.2 How Do Antivirus Software and Privileges Affect Agents?

This section provides the following information about agents:

- [Section 19.2.1, "How Does Antivirus Software Affect Agents?"](#)
- [Section 19.2.2, "What Privileges Affect Agents?"](#)

19.2.1 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 start 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.

19.2.2 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 "Managing Presentation Services Privileges" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

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*.

19.3 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 19.3.1, "Manually Configuring Presentation Services Settings That Affect Agents"](#)
- [Section 19.3.2, "Manually Changing Additional Scheduler Settings That Affect Agents"](#)
- [Section 19.3.3, "What Additional Scheduler Configuration Settings Affect Agents?"](#)
- [Section 19.3.4, "Controlling Delivery Options for Agents"](#)

19.3.1 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 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 19–1](#).
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 19–1 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

Table 19–1 (Cont.) Presentation Services Settings That Affect Agents

Element	Description	Default Value
DefaultDeliveryFormat	<p>Specifies the default format for sending e-mailed reports through an agent.</p> <p>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:</p> <p>html pdf excel text</p>	html

19.3.2 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 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 19.3.3, "What Additional Scheduler Configuration Settings Affect Agents?"](#)
3. Include the elements and their ancestor elements as appropriate. The entry for Log_Dir is shown in the following example:

```
<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.

19.3.3 What Additional Scheduler Configuration Settings Affect Agents?

You can change the following additional Oracle BI Scheduler configuration settings that affect agents:

- [Section 19.3.3.1, "General Scheduler Configuration Settings That Affect Agents"](#)
- [Section 19.3.3.2, "E-Mail Scheduler Configuration Settings That Affect Agents"](#)
- [Section 19.3.3.3, "Agent Scheduler Configuration Settings"](#)

19.3.3.1 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 19–2](#) describes the settings.

Table 19–2 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 "Changing Oracle BI Scheduler Table Names" in <i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i> and see <i>Oracle Business Intelligence Applications Installation and Configuration Guide</i> .	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_ERRMSG	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 <code>ORACLE_INSTANCE\bifoundation\OracleBISchedulerComponent\coreapplication_obischn\scripts\scheduler</code> , for example <code>D:\OBI11g\instances\instance1\bifoundation\OracleBISchedulerComponent\coreapplication_obisch1\scripts\scheduler</code> .	scripts\scheduler

Table 19–2 (Cont.) General Scheduler Configuration Settings That Affect Agents

Element	Description	Default Value
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 <i>ORACLE_INSTANCE\bifoundation\OracleBISchedulerComponent\coreapplication_obischn\scripts\scheduler\common</i> , 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.
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

Table 19–2 (Cont.) General Scheduler Configuration Settings That Affect Agents

Element	Description	Default Value
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
CACertificateFile	Specifies the CA Certificate File. This setting supports SSL.	NA
TrustedPeerDNs	Specifies the SSL Trusted Peer DN's.	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

19.3.3.2 E-Mail Scheduler Configuration Settings That Affect Agents

Table 19–3 E-Mail 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 information, see "Advanced SSL Configuration Options" in <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i> .	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

19.3.3.3 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 19–4 describes each agent configuration element.

Table 19–4 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 about log files, see Section 8.5.6, "Diagnosing Issues with 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
NumDeliveryRetries	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

Table 19–4 (Cont.) Agent Scheduler Configuration Settings

Element	Description	Default Value
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
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	<p>Debug Enabled.</p> <p>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 <code>ORACLE_INSTANCE\diagnostics\logs\OracleBISchedulerComponent\coreapplication_obischn</code>.</p> <p>A new log file named <code>Agent-<Job number>-<Instance number>.log</code> is created for each job instance. The Job Manager can also be used to override the Debug setting for an individual job.</p> <p>For more information, see Section 8.5.6, "Diagnosing Issues with Agents."</p>	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 <code>ORACLE_INSTANCE\diagnostics\logs\OracleBISchedulerComponent\coreapplication_obischn</code> .	true

19.3.4 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 "Setting Up the SA System Subject Area" in *Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition* for information.

19.4 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 about privileges, see "Managing Presentation Services Privileges" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

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 "Administration page" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

19.5 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 "Administration page" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

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 20.1, "Configuring MapViewer to Support Map Views"](#)
- [Section 20.2, "Manually Configuring for Map Views"](#)
- [Section 20.3, "Inserting Text on a Map"](#)
- [Section 20.4, "Configuring Maps for External Consumption"](#)

See [Chapter 13, "Configuring Mapping and Spatial Information"](#) for additional information. 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 documentation library for Oracle Database.

20.1 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 nonproxy 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.

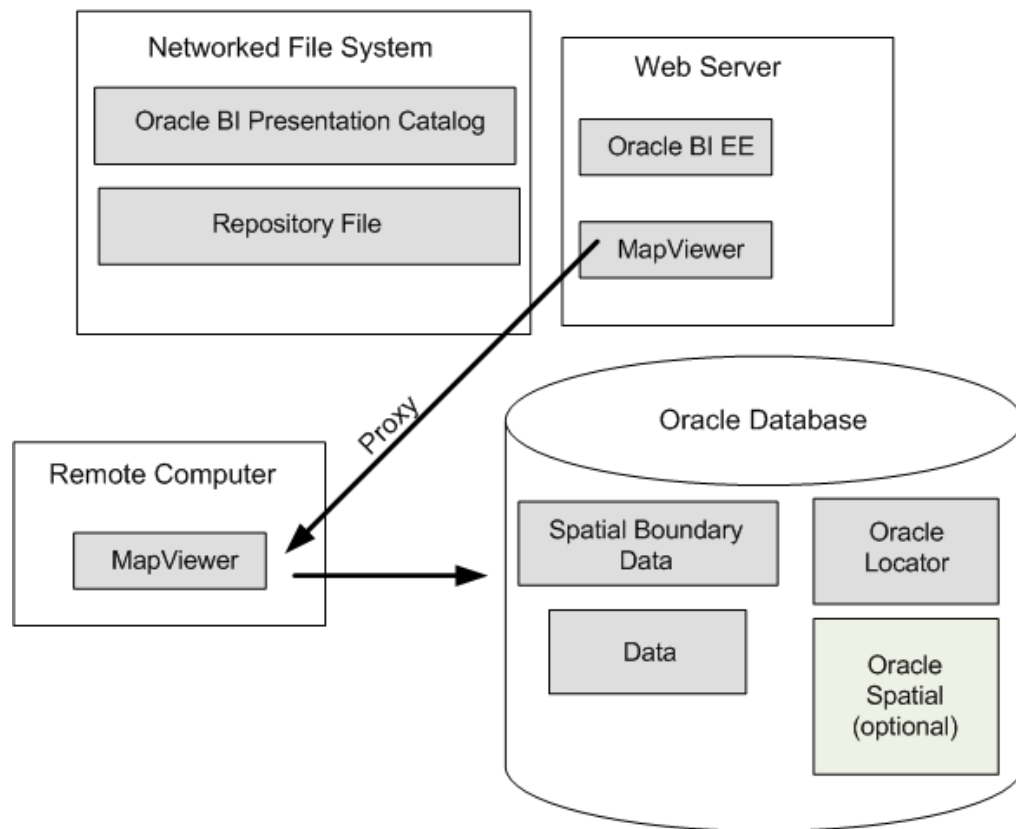
Complete the following steps to configure for a remote MapViewer:

- Edit the RemoteOracleMapViewerAbsoluteURL element in the instanceconfig.xml file, as described in [Section 20.2, "Manually Configuring for Map Views."](#)
- Edit the proxy_enabled_hosts element in the mapViewConfig.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>
```

Figure 20–1 shows the preferred architecture for map views, which provides better performance through a proxy than the default architecture that is shown in Figure 13–1. You can store the data either in an Oracle Database or in other databases that Oracle BI EE supports.

Figure 20–1 Preferred Architecture for Map Views



20.2 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 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:
 - ColocatedOracleMapViewContextPath — Specifies the context path of the MapViewer instance that is co-located with Oracle BI EE. The default value is /mapviewer.
 - RemoteOracleMapViewAbsoluteURL — 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:

```
<RemoteOracleMapViewAbsoluteURL>http://remoteserver:9704/mapviewer
</RemoteOracleMapViewAbsoluteURL>
```

- 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.
 - MaxColumns — Specifies the maximum number of slices and bars generated for pie and bar graphs on the map. The setting applies to all layers on the map and overrides the MaxVisibleColumns element that applies to data cubes. The default value is 300. If the format for a layer causes this value to be exceeded, then an error message is displayed instead of the map. The parent element is LayerDataLayout.
 - SyndicatedOracleMapViewContextPath — Specifies the URL of the MapViewer instance for embedding maps in external pages. For details and an example, see [Section 20.4, "Configuring Maps for External Consumption."](#)
3. Include the elements and their ancestor elements as appropriate, as shown in the following example.

```
<ServerInstance>
  <SpatialMaps>
    <ColocatedOracleMapViewContextPath>/mapviewer</ColocatedOracleMapViewContextPath>
    <RemoteOracleMapViewAbsoluteURL></RemoteOracleMapViewAbsoluteURL>
    <LayerDataLayout>
      <MaxRecords>600</MaxRecords>
      <MaxColumns>400</MaxColumns>
    </LayerDataLayout>
  </SpatialMaps>
</ServerInstance>
```

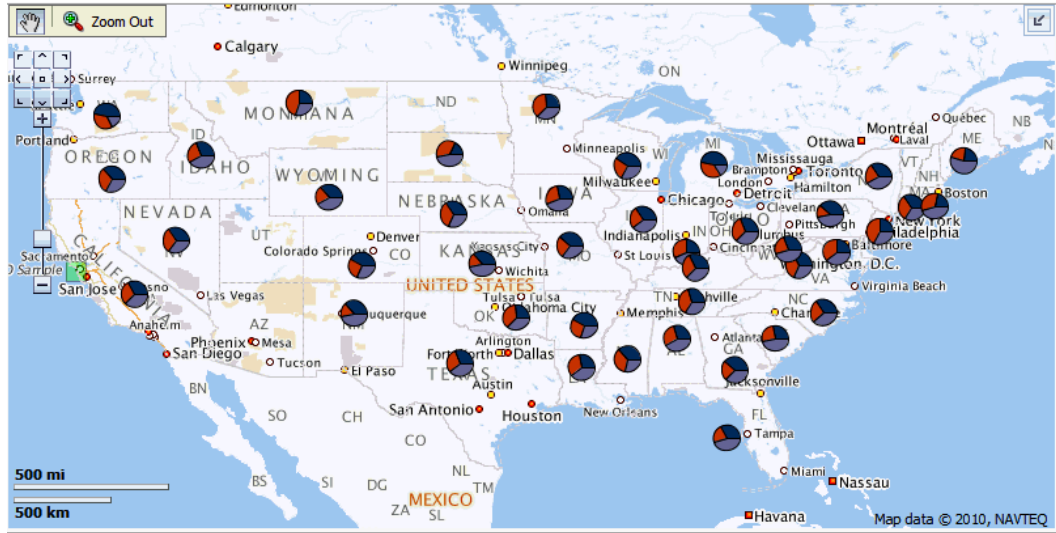
4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

20.3 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 20–2 shows an example of a copyright string on a map. The string is in the lower-right corner.

Figure 20–2 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. On the Edit page, enter the appropriate text in the **Copyright text** field.
For example, the following code displays the copyright line that is shown in Figure 20–2.

```
<copyright>Map data © 2010, NAVTEQ</copyright>
```
4. Click the **Submit** button to save your changes.
5. If you do not see the updated text on the map, then click the browser's **Refresh** button to refresh the map.

20.4 Configuring Maps for External Consumption

You can embed map views in external pages, such as those from Oracle Portal, after setting the appropriate configuration options. 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 `SyndicatedOracleMapViewContextPath` 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>
    <SyndicatedOracleMapViewContextPath>/mapviewer</SyndicatedOracleMapViewContextPath>
    <RemoteOracleMapViewAbsoluteURL>http://myserver:9704/mapviewer</RemoteOracleMapViewAbsoluteURL>
  </SpatialMaps>
</ServerInstance>
```

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 subelements, 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 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 21-1](#).
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 21–1 Elements for URL Generation and Resource File Location

Element	Description	Default Value
AbsoluteCommandURLPrefix	<p>Specifies how Presentation Services generates command URLs. If you explicitly specify an value, then it must be of the following form:</p> <p><i>protocol://server/virtualpath</i></p> <p>where <i>virtualpath</i> 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.</p>	Varies
CustomerResourcePhysicalPath	<p>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 <i>ORACLE_HOME\bifoundation\web\app\res</i>.</p> <p>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.</p>	Varies
CustomerResourceVirtualPath	<p>Specifies the virtual path used for resource files that are not part of a default installation as specified in the CustomerResourcePhysicalPath element.</p>	NA
ForceAbsoluteResourceURL	<p>Specifies whether Presentation Services always generates fully qualified URLs for resource files that have fully qualified virtual paths.</p> <p>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.</p>	False

Table 21-1 (Cont.) Elements for URL Generation and Resource File Location

Element	Description	Default Value
ResourcePhysicalPath	<p>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 <i>ORACLE_HOME\bifoundation\web\app\res</i>.</p> <p>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.</p> <p>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.</p>	NA
ResourceServerPrefix	<p>Specifies how Presentation Services generates URLs for static resources such as images, script files, style sheets, and other user-specified files. The default is <i>protocol://server</i> from the AbsoluteCommandURLPrefix element.</p> <p>If you explicitly specify a value, then it must be of this form:</p> <p><i>protocol://server</i></p> <p>If you specify a virtual path, then it is removed.</p> <p>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.</p>	<i>protocol://server</i>

Table 21-1 (Cont.) Elements for URL Generation and Resource File Location

Element	Description	Default Value
ResourceVirtualPath	<p>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.</p> <p>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.</p> <p>For generating absolute URLs, the value of the AbsoluteCommandURLPrefix element is used as the default.</p> <p>The value must be a fully qualified virtual path of this form:</p> <p><i>' /VirtualPath'</i></p> <p>If you omit the leading slash, then one is added.</p>	res

Part VII

Automating Management of Oracle Business Intelligence

This part explains how to use the Oracle BI Systems Management API. It includes the following chapter:

- [Chapter 22, "Introducing the Oracle BI Systems Management API"](#)

Introducing the Oracle BI Systems Management API

This chapter describes the Oracle BI Systems Management Application Programming Interface (API) and includes the following topics:

- [Section 22.1, "What Is the Oracle BI Systems Management API?"](#)
- [Section 22.2, "Accessing Oracle BI Systems Management API Methods"](#)
- [Section 22.3, "Using the Oracle BI Systems Management API to Manage and Configure Oracle Business Intelligence"](#)
- [Section 22.4, "Descriptions of JMX Admin MBeans and Methods for the Oracle BI Systems Management API"](#)

22.1 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.

22.2 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 22.4, "Descriptions of JMX Admin MBeans and Methods for the Oracle BI Systems Management API."](#)

22.3 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 22.3.1, "Displaying Oracle Business Intelligence Status Using the Oracle BI Systems Management API"](#)
- [Section 22.3.2, "Starting and Stopping Oracle Business Intelligence Using the Oracle BI Systems Management API"](#)
- [Section 22.3.3, "Updating Configuration Settings Using the Oracle BI Systems Management API"](#)
- [Section 22.3.4, "Scaling Out for High Availability and Performance Using the Oracle BI Systems Management API"](#)
- [Section 22.3.5, "Capturing Metrics Using the Oracle BI Systems Management API"](#)

22.3.1 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 22.3.1.1, "Example: Using WLST Commands in a Script to Display the Status of Oracle Business Intelligence"](#)
- [Section 22.3.1.2, "Example: Using a Java Program to Display the Status of Oracle Business Intelligence"](#)

22.3.1.1 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, use the SSL port or
Admin port 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 *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

22.3.1.2 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 22.3.2.2, "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 `\<wls>\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=BIInstance,biInstance=coreapplication,group=Ser
vice

Status is: FULLY_STARTED

```

22.3.2 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 22.3.2.1, "Example: Using WLST Commands to Stop and Start Oracle Business Intelligence"](#)
- [Section 22.3.2.2, "Example: Using a Java Program to Stop and Start Oracle Business Intelligence"](#)

22.3.2.1 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 enter 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, use the SSL port or
Admin port 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.

22.3.2.2 Example: Using a Java Program to Stop and Start Oracle Business Intelligence

Refer to the example in [Section 22.3.1.2, "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();
}

```

22.3.3 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 22.3.3.1, "Example: Using WLST to Edit the E-Mail Sender Display Name Configuration Setting"](#)
- [Section 22.3.3.2, "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 22.4, "Descriptions of JMX Admin MBeans and Methods for the Oracle BI Systems Management API."](#)

22.3.3.1 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, use the SSL port or
Admin port 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.

```

22.3.3.2 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=BIInstance,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 \<wlsserver>\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=BIInstance,biInstance=coreapplication,group=Service
Getting Email MBean ...
Found BI Email MBean:
oracle.biee.admin:type=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
```

22.3.4 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 22.3.4.1, "Example: Using WLST to Scale Out to Servers"](#)
- [Section 22.3.4.2, "Example: Using a Java Program to Scale Out to Servers"](#)

22.3.4.1 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 is needed 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, use the SSL port or
Admin port 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=Service
```

```
BIInstance MBean; ServiceStatus: FULLY_STARTED
```

```
Looping through OracleInstances looking for: instance1
```

```
About to cd to:
```

```
oracle.biee.admin:oracleInstance=instance1,type=BIDomain.OracleInstance,group=Service
```

```
Found instance with id: instance1
```

```
Found matching OracleInstance...
```

```
Acquired BIInstanceDeployment :
```

```
oracle.biee.admin:oracleInstance=instance1,type=BIDomain.OracleInstance.BIInstanceDeployment,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 *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

22.3.4.2 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=BI Domain,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 `<wls>\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 \<wls>\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=BIInstance,biInstance=coreapplication,group=Service
Found Oracle Instance MBean:
oracle.biee.admin:oracleInstance=instance1,type=BIInstance,group=Service
Status is: PARTIALLY_STARTED
Locking domain ...
Locked domain OK
Found BI Instance Deployment MBean:
oracle.biee.admin:oracleInstance=instance1,type=BIInstance,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
```

22.3.5 Capturing Metrics Using the Oracle BI Systems Management API

In addition to the Metrics Browser in Fusion Middleware Control, you can view metrics for Oracle Business Intelligence using the Dynamic Monitoring Service (DMS) and WLST commands. This section describes how to use these methods.

22.3.5.1 Using the Dynamic Monitoring Service for Metrics

You can use the Dynamic Monitoring Service (DMS) to view metrics for Oracle Business Intelligence. Access the service using the following URL:

```
http://<host>:7001/dms
```

Using the Metrics Tables list in the left pane, select **Non-J2EE Metrics** to view the list of metrics for Oracle Business Intelligence. This is the same list that you see in the Metrics Browser of Fusion Middleware Control.

You can use the Dynamic Monitoring Service to quickly obtain a snapshot of metrics. You can use the URL for a given metric as the source for a web query in a Microsoft Excel spreadsheet that you combine with a macro that automatically copies values to an archive sheet and refreshes the query on a loop for a given period.

Suppose that you want to use the Dynamic Monitoring Service to see the details of the metric table called "Oracle_BI_General". When you click the **Oracle_BI_General** link in the Metrics Tables list, the table is displayed on the right side. This metric table consists of several monitored values, such as Active_Execute_Requests and Total_Sessions. You can use the information from the tables that are displayed in this Metrics Browser as part of WLST commands.

For information on accessing DMS metrics using WLST commands, see the chapter on DMS custom WLST commands in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

22.3.5.2 Using WLTS Commands for Metrics

You can use WLST commands to capture metrics for Oracle Business Intelligence.

To use WLST commands for metrics:

1. Navigate to the `MW_HOME/ORACLE_HOME/common/bin` directory.
2. Run the WLST utility.
3. Connect to the Oracle BI system using the connect command, as shown in the following example:

```
connect('user','password','t3://<host>:7001')
```

4. Verify that you are in "online mode" by viewing the following prompt:

```
wls:/bifoundation_domain/serverConfig>
```

You can now interactively use the DMS custom WLST commands. For example, to list all the metric tables that start with "Oracle_BI", enter the following command:

```
wls:/bifoundation_domain/serverConfig>
displayMetricTables('Oracle_BI*')
```

This command generates a long list of data for each of the Oracle BI metric tables. So it is more useful to focus on a given metric table, such as "Oracle_BI_General". The following command displays output such as that shown in this sample.

```
wls:/bifoundation_domain/serverConfig>
displayMetricTables('Oracle_BI_General')
```

```
-----
Oracle_BI_General
-----
Active_Execute_Requests.value: 0
Active_Fetch_Requests.value: 0
Active_File_Handles.value: 1
Active_Initblock_Executions.value: 0
Active_Logins.value: 0
Active_Prepare_Requests.value: 0
Avg._Failed_Logins_Elapsed_Time.value: 0
Avg._Initblock_Executions_Elapsed_Time.value: 0
Avg._Succeeded_Logins_Elapsed_Time.value: 0
Avg._query_elapsed_time.value: 0
Busy_File_Handles.value: 0
File_Handle_Waiters.value: 0
```

```

Free_File_Handles.value:          502
Host:    oracle-bc5ac6af
Max._Initblock_Execution_Elapsed_Time.value:  0
Max_File_Handles.value: 503
Name:    Oracle BI General
New_Execute_Requests.value:      19
New_Fetch_Requests.value:        32
New_Initblock_Executions.value:  0
New_Logins.value:                 7
New_Prepare_Requests.value:      19
New_Requests.value:              187
OBPERF_***.value:                 7
Oracle_BI_Applications: Oracle BI Server
Parent:  /Oracle BI Server
Process: Oracle BI Server:4004:/instance1/coreapplication_obis1
Queries/sec.value:                0
ServerName:    /instance1/coreapplication_obis1
Succeeded_Initblock_Execution_Ratio_as_%.value: 0
Succeeded_Logins_Ratio_as_%.value:    7
Total_sessions.value:              0

```

Using the scripting capability of WLST, you can embed DMS commands into a Jython script to store the required metric values in a file. The following is an example of such a script.

```

# Script to dump timestamp (in milliseconds) for a single Oracle BI metric
# to a file
#
from java.util import Date
from java.text import SimpleDateFormat
#
# Modify to connect to your server
connect('biadmin','welcome1','t3://localhost:7001')
#
# This is the number of times to sample the metric
sample_length = 100
#
# This is where you define what metric table and metric to dump to file
metric_table = "Oracle_BI_General"
metric_of_interest = "Avg._query_elapsed_time.value"
#
# Some metrics have non-text characters in the name. Provide a reference here
# so it dumps to file without error in file name
output_file_metric_ref = "Avg_Qry_Elapse"
#
# This section defines the output file name with unique time
start_time = str(SimpleDateFormat("dd-MMM-yyyy_HH-mm-ss").format(Date()))
output_filename = start_time + "_" + output_file_metric_ref + "_dump.txt"
#
# Open the file and write summary of metric to be dumped
file = open(output_filename,'w')
print >>file, "Start Metric Dump of: " + str(metric_table) + " : " + str(metric_
of_interest) + " at " + str(SimpleDateFormat("dd-MMM-yyyy
HH-mm-ss").format(Date()))
#
#
# The following section forms a loop according to the sample length defined
# earlier. The 'displayMetricTables()' command returns the metric table in the
# form of a JMX composite data array. The code following this command access
# the metric data from this array. In this case, a particular metric of

```

```

# interest is tested for and only the value of that metric is output to file.
#
counter = 0
while counter <= sample_length:
    results = displayMetricTables(metric_table)
    for table in results:
        name = table.get('Table')
        rows = table.get('Rows')
        rowCollection = rows.values()
        iter = rowCollection.iterator()
        while iter.hasNext():
            row = iter.next()
            rowType = row.getCompositeType()
            keys = rowType.keySet()
            keyIter = keys.iterator()
            while keyIter.hasNext():
                columnName = keyIter.next()
                value = row.get(columnName)
                if columnName == metric_of_interest:
                    print >>file, str(SimpleDateFormat("dd-MMM-yyyy
HH-mm-ss-SSS").format(Date())) + "," + str(value)
            counter = counter + 1
file.close()
disconnect()

```

Certain Oracle BI metric tables, such as "Oracle_BI_Thread_Pool", are in effect two-dimensional. With the "Oracle_BI_Thread_Pool" table, you can query the metric values for various "Names", such as "Server" or "Usage_Tracking". To export the required metric value to a file in this case, you must modify the logic that was used in looping in the previous example script to handle the two dimensions. The following example script provides one way to handle this case.

```

# Script to dump timestamp (in milliseconds) and a
#single Oracle BI metric to a file for metrics with multiple sections
#
from java.util import Date
from java.text import SimpleDateFormat
#
# Modify to connect to your server
connect('biadmin','welcome1','t3://localhost:7001')
#
# This is the number of times to sample the metric
sample_length = 100
#
# This is where you define what metric table, category, and metric to
# dump to file
metric_table = "Oracle_BI_Thread_Pool"
category_of_interest = "Server"
metric_of_interest = "Avg._Request/sec.value"
#
# Some metrics have non-text characters - provide a reference here
# so it dumps to file without error
output_file_metric_ref = "Avg_Req_Sec"
#
# This section defines the output file name with unique time
start_time = str(SimpleDateFormat("dd-MMM-yyyy_HH-mm-ss").format(Date()))
output_filename = start_time + "_" + output_file_metric_ref + "_dump.txt"
#
# Open the file and write summary of metric to be dumped
file = open(output_filename,'w')

```



```

print >>file, "Start Metric Dump of: " + str(metric_table) + " : " + str(metric_
of_interest) + " for Category: " + str(category_of_interest) + " at " +
str(SimpleDateFormat("dd-MMM-yyyy HH-mm-ss").format(Date()))
#
#
counter = 0
while counter <= sample_length:
    results = displayMetricTables(metric_table)
    for table in results:
        name = table.get('Table')
        rows = table.get('Rows')
    rowCollection = rows.values()
    iter = rowCollection.iterator()
    while iter.hasNext():
        row = iter.next()
        if row.containsValue(category_of_interest):
            rowType = row.getCompositeType()
            keys = rowType.keySet()
            keyIter = keys.iterator()
            while keyIter.hasNext():
                columnName = keyIter.next()
                value = row.get(columnName)
                if columnName == metric_of_interest:
                    print >>file, str(SimpleDateFormat("dd-MMM-yyyy
HH-mm-ss-SSS").format(Date())) + ", " + str(value)
            counter = counter + 1
file.close()
disconnect()
    
```

22.4 Descriptions of JMX Admin MBeans and Methods for the 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 22–1](#) lists the JMX Admin MBeans for the Oracle BI Systems Management API.

Table 22–1 Oracle BI Systems Management API JMX Admin MBeans

Admin MBean	Description
<code>AvailabilityConfigurationMBean</code>	Provides an MBean interface for configuring several options that affect process availability for Oracle Business Intelligence.
<code>BICComponentMBean</code>	Defines an MBean interface for a component of a BI domain.
<code>BIDomainMBean</code>	Defines an MBean interface for managing a BI domain.

Table 22–1 (Cont.) Oracle BI Systems Management API JMX Admin MBeans

Admin MBean	Description
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 Presentation Services.
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 VIII

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 about installation, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

For information about upgrade, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*.

This part includes the following chapters on life-cycle management tasks:

- [Chapter 23, "Patching Oracle Business Intelligence Systems"](#)
- [Chapter 24, "Moving Between Environments"](#)
- [Chapter 25, "Backup and Recovery of Oracle Business Intelligence Systems"](#)

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 *Oracle Fusion Middleware Patching Guide*.

This chapter includes the following topics:

- [Section 23.1, "What Is Patched for the Oracle Business Intelligence Platform?"](#)
- [Section 23.2, "Patching the Platform"](#)
- [Section 23.3, "Rolling Back a Platform Patch"](#)
- [Section 23.4, "Determining Current Patch Levels"](#)

23.1 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

Nondesign-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.

23.2 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 Services, as described in [Section 23.2.2, "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 23.2.1, "What Conditions Apply when Patching the Platform?"](#)
- [Section 23.2.2, "Applying a Patch to the Platform"](#)
- [Section 23.2.3, "What Happens If a Patching Conflict Occurs?"](#)

23.2.1 What Conditions Apply when Patching the Platform?

The following conditions apply when patching the Oracle BI EE platform:

- Oracle BI EE 11g Release 1 (11.1.1) must be installed, and previous versions must have been upgraded to this 11g 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.

23.2.2 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 23.4, "Determining Current Patch Levels."](#)

Complete the tasks that are described in the following sections to apply a patch:

1. [Section 23.2.2.1, "Step 1: Download the Patch"](#)
2. [Section 23.2.2.2, "Step 2: Stop System Component Processes for Oracle BI EE"](#)
3. [Section 23.2.2.3, "Step 3: Apply the Patch to Each Oracle Home"](#)
4. [Section 23.2.2.4, "Step 4: Start System Component Processes for Oracle BI EE"](#)
5. [Section 23.2.2.5, "Step 5: Redeploy Patched J2EE Applications"](#)

23.2.2.1 Step 1: Download the Patch

To download patch:

1. Locate the patch (OPatch package) on the Oracle Support Services 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 Services.

2. Download the patch to your local network.

For more information, see *Oracle Fusion Middleware Patching Guide*.

23.2.2.2 Step 2: Stop System Component Processes for Oracle BI EE

For details, see [Section 4.1, "About Starting and Stopping Oracle Business Intelligence."](#)

23.2.2.3 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

23.2.2.4 Step 4: Start System Component Processes for Oracle BI EE

For details, see [Section 4.1, "About Starting and Stopping Oracle Business Intelligence."](#)

23.2.2.5 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 *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 restarted 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.

23.2.3 What Happens If a Patching Conflict Occurs?

If a patching conflict occurs, then the process stops and you should contact Oracle Support Services.

23.3 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 23.4, "Determining Current Patch Levels."](#)

Complete the tasks that are described in the following sections to apply a patch:

- [Section 23.3.1, "Step 1: Stop System Component Processes for Oracle BI EE"](#)
- [Section 23.3.2, "Step 2: Invoke OPatch with "-rollback" Option"](#)
- [Section 23.3.3, "Step 3: Start System Component Processes for Oracle BI EE"](#)
- [Section 23.3.4, "Step 4: Redeploy J2EE Applications"](#)

23.3.1 Step 1: Stop System Component Processes for Oracle BI EE

For details, see [Section 4.1, "About Starting and Stopping Oracle Business Intelligence."](#)

23.3.2 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

23.3.3 Step 3: Start System Component Processes for Oracle BI EE

For details, see [Section 4.1, "About Starting and Stopping Oracle Business Intelligence."](#)

23.3.4 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 *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.

23.4 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*.

Moving Between Environments

You can move Oracle Business Intelligence to a new environment or from a test to a production environment, as described in the following list:

- 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.

Moving is sometimes performed after a system has been upgraded. Upgrade is not covered in this guide. For complete information, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*.

- 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 on the same operating system. This can also include moving from a single- to a multiple-computer environment.

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.

For information, see "Moving Oracle Business Intelligence to a Production Environment" in *Oracle Fusion Middleware Administrator's Guide*. The `pasteBinary` script is used when moving to production environments. This script is not supported with Oracle Business Intelligence installations that were installed on 64-bit operating systems using the Enterprise Installation type.

In addition, Oracle Business Intelligence supports whole server migration, in which a WebLogic Server instance is migrated to a different physical computer upon failure, either manually or automatically. For complete information, see the chapter on "Whole Server Migration" in *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*.



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 "Backup and Recovery Recommendations for Oracle Business Intelligence" in *Oracle Fusion Middleware Administrator's Guide*.



Part IX

Reference Information

This part provides reference information for managing Oracle Business Intelligence. It includes the following appendixes:

- [Appendix A, "NQSCONFIG.INI File Configuration Settings"](#)
- [Appendix B, "Advanced Configuration Reference"](#)
- [Appendix C, "Mapping User Interface Labels with Configuration File Elements"](#)
- [Appendix D, "Manually Updating Fusion Middleware Control Settings"](#)

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 includes 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"](#)
- [Section A.13, "Datamart Automation 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 C, "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 NQSSConfig.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 NQSSConfig.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 10.2, "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 NQSSConfig.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;*

In this syntax:

- *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 quotation marks.
- *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 NQSSConfig.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 quotation marks (") 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. Therefore, the DATA_STORAGE_PATHS entry must be unique for each clustered server. To ensure this unique entry, enter a relative path so that the 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) occurs 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 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.

- Noncacheable 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.
- Noncacheable table. Physical tables in the Oracle BI Server repository can be marked "noncacheable." If a query references any noncacheable 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 8.4.1, "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, you might disable subrequest caching, 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 `NQConfig.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 install additional content on the 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 15, "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.
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 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 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 15.4.1, "Setting Locale Parameters on the 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 about how this parameter relates to the case setting in Oracle BI Presentation Services, see [Section B.1, "Making Advanced Configuration Changes for 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 `NQSCfg.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 quotation marks (") 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 occurs 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 larger 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 larger 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 YES.

Example: PROJECT_INACCESSIBLE_COLUMN_AS_NULL = YES;

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 Sockets 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 file name. For client components, such as Oracle Business Intelligence ODBC Client Data Source, this is the Client Certificate file name.

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 a trusted CA. A depth of two means the certificate was signed by a CA that was further verified by a CA. 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 `NQSSConfig.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 server threads up to the specified maximum using the parameter `SERVER_THREAD_RANGE`. All the threads that are 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 larger 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 larger 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, according to `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 failure. 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 Services.

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 Services.

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 Services.

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 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 `NQSCfg.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 `NQSSConfig.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 `ClusterConfig.xml` file in the following location:

`ORACLE_INSTANCE/config/OracleBIApplication/coreapplication`

For more information, see the information about the `ClusterConfig.xml` 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 10.2, "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 quotation marks (") 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 (the one for which the MasterServer parameter is set to true in the ClusterConfig.xml file) must have read and write access to this directory. The Oracle BI Servers in the cluster (the other servers defined in the ClusterConfig.xml 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 `NQSSConfig.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 10.2, "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 DISCONNECTED

This parameter has been deprecated and is no longer used.

A.6.21 AUTOMATIC_RESTART

Specifies whether the Oracle BI Server should be automatically restarted after a failure. 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.6.22 VARIABLE_VALUE_LIMIT

Specifies the maximum length of returned session variable values when client tools call the `NQSGetSessionValues()` function.

Example: `VARIABLE_VALUE_LIMIT= 10;`

For example, suppose `VARIABLE_VALUE_LIMIT` is set to 10 and the `NQSGetSessionValues()` function is called on a variable whose value is "1234567890ABCDE." The value is truncated to "1234567890".

A.6.23 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.6.24 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.6.25 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.6.26 FMW_UPDATE_ROLE_AND_USER_REF_GUIDS

Users are identified by their global unique identifiers (GUIDs), not by their names. Because of this identification, 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 Security Using the Default Security Configuration" 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;*

In this syntax:

- *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 Services.

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;
- BAPI = nqsdbgatewaysapbapi;
- ESSBASE = nqsdbgatewayessbaseapi;
- OracleADF = nqsdbgatewayoracleadf;
- OracleADF_HTTP = nqsdbgatewayoracleadf;
- HyperionADM = nqsdbgatewayadm;
- OracleWS = nqsdbgatewayoraclews;

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 standalone 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.

Table A-2 (Cont.) Names and Descriptions of Columns Added to Usage Tracking Table

Name	Data Type	Description	Notes
PRESENTATION_NAME	Varchar(128)	Name of the Presentation Catalog in Oracle BI Presentation Services	NA
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 nonzero. 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

Table A-3 (Cont.) Valid Parameters for DIRECT_INSERT Settings

DIRECT_INSERT Setting	Parameters Used	Parameter Setting
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 quotation marks (") 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 to a larger number increases the amount of data that might be lost if the server shuts down abnormally. 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 quotation marks. 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_SCHEDULE"."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 started 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

Specifies the name of the Analytic Workspace object that is created in the target Oracle Database. The aggregate AW cubes and dimensions are created under this container.

Example: `AGGREGATE_AW_NAME = "OBI_AW";`

A.11.4 PREAGGREGATE_AW_CUBE

Specifies whether the system-generated AW cube for aggregate persistence must be fully solved. The default value is YES. Note that a YES value significantly increases storage space usage.

Example: `PREAGGREGATE_AW_CUBE = YES;`

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";`

A.13 Datamart Automation Section Parameters

There is only one parameter in this section. It provides information about Essbase Studio.

A.13.1 ESSBASE_STUDIO_URL

This parameter is reserved for a future release.

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 includes the following topics:

- [Section B.1, "Making Advanced Configuration Changes for Presentation Services"](#)
- [Section B.2, "Using the JavaHost Service for Oracle BI Presentation Services"](#)

B.1 Making Advanced Configuration Changes for 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 can use the `instanceconfig.xml` file to customize various aspects of your deployment. 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.

The following procedure provides information about 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 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 B-1](#).
3. Include the elements and their ancestor elements as appropriate, as shown in the following example:

```
<ServerInstance>
  <UnaccessedRunningTimeoutMinutes>5</UnaccessedRunningTimeoutMinutes>
  <Security>
    <AllowRememberPassword>False</AllowRememberPassword>
    <CookieDomain>value</CookieDomain>
    <CookiePath>/analytics</CookiePath>
    <InIFrameRenderingMode>prohibit</InIFrameRenderingMode>
  </Security>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

Table B-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
CookieDomain	Specifies domain information for a cookie that is sent to the browser.	NA
CookiePath	Specifies the domain path to which cookies apply.	/analytics
InIFrameRenderingMode	See Section B.1.1, "Protecting Pages in Oracle BI EE from Attack" for information.	sameDomainOnly
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

B.1.1 Protecting Pages in Oracle BI EE from Attack

As the administrator, you must be aware of a security concern that is known as *clickjacking*. Clickjacking refers to the ability of attackers to subvert clicks and send the victim's clicks to web pages that allow themselves to be framed with or with JavaScript.

For example, suppose an attacker develops a Web site that uses an inline frame for an Oracle Business Intelligence Console application. When you visit this site, you are unknowingly clicking buttons on the inline-framed Console application. This

vulnerability is very serious, because the attacker is not stopped by the same origin policy principles that apply to other Oracle Business Intelligence applications. You can find many examples of clickjacking documented on the Worldwide Web.

The term that describes preventing attackers from framing an application in an inline frame is *frame busting*. To affect frame busting, you use the `InIFrameRenderingMode` element in the `instanceconfig.xml` file. You can set the element to the following three values:

- `prohibit` = Never allow content from Oracle BI Presentation Services to be rendered in an inline frame.
- `sameDomainOnly` = (Default) Allow rendering of pages in an inline frame if the enclosing page was generated by the server in the same domain. By default, pages have the same domain if they were generated by the same server. See the Worldwide Web for information on the "same origin policy."
- `allow` = Always allow content from Oracle BI Presentation Services to be rendered in an inline frame.

B.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_objjhn` directory. See [Table B-2](#) for a description of the elements. The elements are identified by their relative path starting from the JavaHost element.

The common subelements, such as `InputStreamLimitInKB`, do not apply to the `MessageProcessor`, `Listener`, or `SSL` loaders.

For information about manual editing, see [Section 3.4, "Using a Text Editor to Update Configuration Settings."](#)

Table B-2 JavaHost Service Configuration Elements in `config.xml`

Element	Description
Loaders	Contains the <code>ListOfEnabledLoaders</code> and <code>Loader</code> elements. These elements specify the components for the JavaHost service. Avoid editing the elements in the Loaders section.

Table B-2 (Cont.) JavaHost Service Configuration Elements in config.xml

Element	Description
Loaders/ListOfEnabledLoaders	<p>Specifies the list of components (such as Oracle BI Scheduler and BI Publisher) to be enabled.</p> <p>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.</p> <p>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.</p>
Loaders/Loader	<p>Contains the following elements, which specify configuration information for a specific component:</p> <ul style="list-style-type: none"> ■ 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.
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	<p>A subelement 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.</p> <p>Default: 8192</p>
RequestResponseLogDirectory	<p>A subelement common to each loader that specifies the name of the directory for the response files of requests.</p> <p>Default: A default temp directory</p>
LogLargeRequests	<p>A subelement common to each loader that specifies whether to create a response file when processing large requests.</p> <p>Default: True</p>
ReadRequestBeforeProcessing	<p>A subelement common to each loader that specifies whether to wait to process the request until a file is completely read.</p> <p>Default: True</p>
LargeRequestThresholdInKB	<p>A subelement 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.</p> <p>Default: 200</p>

Table B–2 (Cont.) JavaHost Service Configuration Elements in config.xml

Element	Description
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: <ul style="list-style-type: none"> ■ 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: *
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: <ul style="list-style-type: none"> ■ Yes: Enables SSL encryption ■ No: Disables SSL encryption Default: No For information about SSL, see <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i> .
Batik	Contains only the common subelements such as InputStreamLimitInKB, as they relate to converting SVG images to rasterized image formats.
Scheduler	Contains the following elements: <ul style="list-style-type: none"> ■ Enabled ■ DefaultUserJarFilePath ■ DefaultTempFilePath ■ DefaultPurgingPeriod
Scheduler/Enabled	Specifies whether to enable the interaction of the JavaHost service with Oracle BI Scheduler: <ul style="list-style-type: none"> ■ True: Enables interaction with Oracle BI Scheduler ■ False: Disables interaction with Oracle BI Scheduler Default: False

Table B-2 (Cont.) JavaHost Service Configuration Elements in config.xml

Element	Description
Scheduler/DefaultUserJarFilePath	<p>Specifies the default directory for storing JAR files for the Java extension utility.</p> <p>When Oracle BI Scheduler is enabled, this element is required and accepts a single path.</p>
Scheduler/DefaultTempFilePath	<p>Specifies the default directory for storing temporary files for Oracle BI Scheduler requests.</p> <p>Default: the system temp directory</p>
Scheduler/DefaultPurgingPeriod	<p>Specifies the default period (in seconds) for Oracle BI Scheduler requests to remove failed jobs.</p> <p>Default: 300</p>
XMLP	<p>Contains only the common subelements such as <code>InputStreamLimitInKB</code> and <code>ReadRequestBeforeProcessing</code>, as they relate to Oracle BI Publisher.</p>
URLConnect	<p>Contains elements that relate to SSL. Avoid modifying these elements.</p>
DVT	<p>Contains only the common <code>InputStreamLimitInKB</code> subelement by default, as they relate to graph generation. You can add other common subelements as necessary.</p>

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.

- [Table C-1, " Scalability Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-2, " Availability Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-3, " Performance Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-4, " Log Configuration Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-5, " Security Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-6, " Presentation Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-7, " Repository Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-8, " Scheduler Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-9, " Marketing Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)
- [Table C-10, " Mail Tab - Mapping Between User Interface Labels and Configuration File Elements"](#)

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 C-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	<ul style="list-style-type: none"> ▪ CLUSTER_PARTICIPANT ▪ All elements under the Node element, where NodeType is set to Server 	<ul style="list-style-type: none"> ▪ NQSConfig.INI ▪ ClusterConfig.xml 	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	<ul style="list-style-type: none"> ▪ JavaHostProxy ▪ Java_Host_Server ▪ JAVAHOST_HOSTNAME_OR_IP_ADDRESSES 	<ul style="list-style-type: none"> ▪ instanceconfig.xml (for Presentation Services) ▪ instanceconfig.xml (for Oracle BI Scheduler) ▪ NQSConfig.INI 	Section 5.5
Port Range From / Port Range To	<ul style="list-style-type: none"> ▪ RPC_SERVICE_OR_PORT ▪ JAVAHOST_HOSTNAME_OR_IP_ADDRESSES ▪ Listener ▪ Web_Server PortString ClusterPort ▪ Port 	<ul style="list-style-type: none"> ▪ 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.	Not Available	Chapter 5

Table C-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	<ul style="list-style-type: none"> ▪ All elements under the Node element, where NodeType is set to PrimaryController or Scheduler ▪ Alerts 	<ul style="list-style-type: none"> ▪ ClusterConfig.xml ▪ instanceconfig.xml (for Presentation Services) 	Section 6.2.1, "Using Fusion Middleware Control to Configure Primary and Secondary Instances"
Secondary Host / Instance	<ul style="list-style-type: none"> ▪ All elements under the Node element, where NodeType is set to SecondaryController or Scheduler ▪ Alerts 	<ul style="list-style-type: none"> ▪ ClusterConfig.xml ▪ instanceconfig.xml (for Presentation Services) 	Section 6.2.1

Table C-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 when Rendering 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	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	DefaultRowsDisplayedInDelivery	instanceconfig.xml	Section 7.3.3, "Using Fusion Middleware Control to Set Configuration Options for Data in Tables and Pivot Tables"

Table C-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 8.2.2.1, "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 8.2.2.1
Maximum File Size Note: Field in Query Logs region.	MaximumFileSizeKb	logconfig.xml (for the BI Server)	Section 8.2.2.1
Maximum Log Age Note: Field in Query Logs region.	MaximumLogAgeDay	logconfig.xml (for the BI Server)	Section 8.2.2.1
Incident Error	IncidentError	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.2.2.1
Error	Error	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.2.2.1

Table C-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 8.2.2.1
Notification	Notification	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.2.2.1
Trace	Trace	instanceconfig.xml (for Presentation Services and Scheduler) logging_config.xml (for JavaHost) ccslogging.xml (for Cluster Controller)	Section 8.2.2.1

For information about diagnostic log configuration files (for example, logconfig.xml), see:

- [Section 8.3.2, "What Are Diagnostic Log Configuration Files and Where Are They Located?"](#)
- [Section 8.3.3, "What Are Log File Message Categories and Levels?"](#)
- [Section 8.3.4, "What is Log File Rotation?"](#)

Table C-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	<i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>
SSO Provider	EnabledSchemas (indirectly associated)	instanceconfig.xml	<i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>

Table C-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 C-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: <ol style="list-style-type: none"> In Fusion Middleware Control, under the WebLogic Domain node, select bifoundation_domain. From the WebLogic Domain menu, select Security and then select Credentials. On the Credentials page, expand oracle.bi.enterprise and select an entry of the format <i>repository.rpd_name</i>. 	Not Available	Section 10.2, "Using Fusion Middleware Control to Upload a Repository and Set the Oracle BI Presentation Catalog Location" <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>
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: <i>logical_name = repository_name.rpd, DEFAULT;</i> For example, Star = SampleAppLite.rpd, DEFAULT; where Star is the DSN, and SampleAppLite.rpd is the name of the RPD file.	NQSConfig.INI	Section 10.2

Table C-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 10.2
Shared Location	REPOSITORY_PUBLISHING_DIRECTORY	NQSConfig.INI	Section 10.2
Catalog Location	CatalogPath	instanceconfig.xml	Section 10.2

Table C-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	"Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler" in <i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i>
Call Interface	CLIType	instanceconfig.xml	<i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i>
Data Source	DSN	instanceconfig.xml	<i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i>
Username	scheduler.schema	Credential found in oracle.bi.enterprise credential map.	<i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i> and <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>
Password	See Username.	Not Available	<i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i>
Confirm Password	See Username.	Not Available	<i>Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition</i>

Table C–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 11.4.1, "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 11.4.1 <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>
Password	See Username.	Not Available	Section 11.4.1
Confirm Password	See Username.	Not Available	Section 11.4.1

Table C–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 <i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>
Password	See Username, in the preceding row.	Not Available	Section 11.1.1
Confirm password	See Username, in the preceding row.	Not Available	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 SmtptCACertificateDir or SmtptCACertificateFile	instanceconfig.xml	Section 11.1.1
CA certificate directory	SmtptCACertificateDir	instanceconfig.xml	Section 11.1.1

Table C-10 (Cont.) 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
CA certificate file	SmtptCACertificateFile	instanceconfig.xml	Section 11.1.1
SSL certificate verification depth	SmtptCertificateVerificationDepth	instanceconfig.xml	Section 11.1.1
SSL cipher list	SmtptCipherList	instanceconfig.xml	Section 11.1.1

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 includes the following topics:

- [Section D.1, "Disabling Fusion Middleware Control from Managing Oracle Business Intelligence Configuration Settings"](#)
- [Section D.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."](#)

D.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 reenabling the element, set it to `true`.
 - c. Save the `biee-domain.xml` file.
3. Restart Oracle Business Intelligence.

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

Administration Server

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

See also [Fusion Middleware Control](#), [Java components](#) and [Managed Server](#).

Administration Tool

See [Oracle BI Administration Tool](#).

agent

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

Agents can dynamically detect information-based problems and opportunities, determine the appropriate individuals to notify, and deliver information to them through a wide range of devices (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. You can use alias tables to set up multiple tables, each with different keys, names, or joins, when a single physical table must serve in different roles. Because alias table names are included in physical SQL queries, you can also use alias tables to provide meaningful table names, making the SQL statements easier to read.

analysis

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

See also [filter](#) and [selection step](#).

analysis criteria

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

See also [analysis](#).

analysis prompt

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

See [dashboard prompt](#) and [prompt](#).

attribute

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

attribute column

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

See [hierarchical column](#).

BI Composer

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

BI domain

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

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

See also [BI instance](#).

BI instance

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

See also [BI domain](#).

BI object

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

BI Search

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

bookmark link

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

See [prompted link](#).

bridge table

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

briefing book

See [Oracle BI Briefing Books](#).

business model

Contains the business model definitions and the mappings from logical to physical tables. Business models are always dimensional, unlike objects in the Physical layer,

which reflect the organization of the data sources. Each business model contains logical tables, columns, and joins.

Business Model and Mapping layer

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

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

business owner

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

catalog

See [Oracle BI Presentation Catalog](#).

cause & effect map

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

See also [Oracle Scorecard and Strategy Management](#).

chronological key

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

Cluster Controller

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

column

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

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

See also [analysis](#), [attribute column](#), [hierarchical column](#), and [measure column](#).

column filter

See [filter](#).

column prompt

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

See also [prompt](#).

complex join

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

condition

Objects that return a single Boolean value based on the evaluation of an analysis or of a key performance indicator (KPI). You use conditions to determine whether agents deliver their content and execute their actions, whether actions links are displayed in dashboard pages, or whether sections and their content are displayed in dashboard pages.

See also [action](#), [action link](#), [agent](#) and [key performance indicator \(KPI\)](#).

connection pool

An object in the Physical layer of an Oracle BI repository that contains the connection information for a data source.

See also [Physical layer](#).

content designer

The user who creates business intelligence objects such as analyses, dashboards, and scorecards.

contextual event action

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

See also [action](#), [Action Framework](#), and [action link](#).

criteria

See [analysis criteria](#).

cube

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

currency prompt

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

See also [prompt](#).

custom view

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

See also [Oracle Scorecard and Strategy Management](#).

dashboard

An object that provides personalized views of corporate and external information. A dashboard consists of one or more pages. Pages can display anything that you can

access or open with a Web browser, such as results of analyses, images, alerts from agents, and so on.

dashboard prompt

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

See [analysis prompt](#) and [prompt](#).

Dashboard URL

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

data source name (DSN)

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

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

database hint

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

dimension

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

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

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

See also [hierarchy](#).

dimension table

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

See also [fact table](#).

driving table

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

DSN

See [data source name \(DSN\)](#).

event polling table

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

Essbase

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

fact table

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

See also [dimension table](#).

filter

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

See also [prompt](#) and [selection step](#).

foreign key

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

fragmentation content

The portion, or fragment, of the set of data specified in a logical table source when the logical table source does not contain the entire set of data at a given level.

Fragmentation content is defined by the logical columns that are entered in the **Fragmentation content** box in the Content tab of the Logical Table Source dialog box.

Fusion Middleware Control

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

global header

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

Go URL

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

hierarchical column

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

See also [attribute column](#).

hierarchy

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

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

See also [dimension](#), [logical level](#), and [presentation hierarchy](#).

hierarchy level

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

See also [presentation level](#).

home page

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

image prompt

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

See also [prompt](#).

initialization block

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

initiative

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

See also [objective](#) and [Oracle Scorecard and Strategy Management](#).

Java components

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

See also [Node Manager](#).

key performance indicator (KPI)

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

See also [Oracle Scorecard and Strategy Management](#).

KPI watchlist

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

See also [key performance indicator \(KPI\)](#).

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, and to enable aggregate navigation and fragmentation.

Managed Server

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

See also [Administration Server](#) and [Fusion Middleware Control](#).

measure column

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

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

metadata

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

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

metadata dictionary

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

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 nonrelational databases. Database applications can use ODBC to access data stored in different types of database management systems, even if each database uses a different data storage format and programming interface.

OPMN

See [Oracle Process Manager and Notification Server \(OPMN\)](#).

Oracle BI Administration Tool

A Windows application that is used to create and edit Oracle BI repositories. The Administration Tool provides a graphical representation of the three parts of a repository: the Physical layer, the Business Model and Mapping layer, and the Presentation layer.

Oracle BI Briefing Books

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

Oracle BI JavaHost

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

Oracle BI Logical SQL View Object

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

Oracle BI Presentation Catalog

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

Oracle BI Presentation Services

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

See also [ODBC](#); [Oracle BI Server](#); [Oracle BI Presentation Catalog](#); [Oracle BI Presentation Services 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. You can use this XML file version of the repository to programmatically modify the metadata. The Oracle BI Server XML API objects correspond to metadata repository objects in an RPD file. These objects differ from XML objects in the Oracle BI Presentation Catalog.

Oracle Business Intelligence Mobile

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

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

Oracle Business Intelligence Web Services

See [Oracle Business Intelligence Session-Based Web Services](#) and [Oracle Business Intelligence Web Services for SOA](#).

Oracle Business Intelligence Session-Based Web Services

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

See also [Oracle Business Intelligence Web Services for SOA](#).

Oracle Business Intelligence Web Services for SOA

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

See also [Oracle Business Intelligence Session-Based Web Services](#).

Oracle Call Interface (OCI)

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

Oracle OLAP

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

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

Oracle Process Manager and Notification Server (OPMN)

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

See also [system components](#).

Oracle Scorecard and Strategy Management

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

Oracle Technology Network (OTN)

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

parent-child hierarchy

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

See also [dimension](#).

parent-child relationship table

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

pass-through calculation

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

permissions

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

See [object properties](#).

perspective

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

See also [initiative](#), [key performance indicator \(KPI\)](#), [objective](#), and [Oracle Scorecard and Strategy Management](#).

physical catalog

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

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

See also [column prompt](#), [currency prompt](#), [filter](#), [image prompt](#), and [variable prompt](#).

prompted link

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

See [bookmark link](#).

query

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

query cache

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

ragged hierarchy

See [unbalanced hierarchy](#).

report

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

See also [analysis](#).

repository

See [Oracle BI repository](#).

repository variable

See [variable](#).

results

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

See also [analysis](#).

scorecard

See [Oracle Scorecard and Strategy Management](#).

selection step

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

See also [analysis](#) and [filter](#).

session variable

See [variable](#).

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 might 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

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

See also [prompt](#).

virtual physical table

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

vision statement

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

See also [mission statement](#) and [Oracle Scorecard and Strategy Management](#).

WebLogic server domain

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

WebLogic Scripting Tool (WLST)

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

XML API

See [Oracle BI Server XML API](#).

A

action framework, 11-2
adapters, relational data source, 1-11
Admin MBeans, 1-7
Administration Server
 confirming whether running, 4-9
 defined, 1-7
 starting and stopping, 4-2, 4-9
Administration Tool
 using to manage clustered system components, 6-8
 using to purge cache, 7-27
agents
 antivirus software, 19-2
 configuring e-mail settings for, 11-1
 definition of, 19-1
 delivery options for, 19-9
 e-mail Scheduler configuration settings, 19-7
 general Scheduler configuration settings, 19-5
 KeepErrorLogFiles configuration element, 19-9
 log directory, viewing entries, 8-21
 log file location, 19-8
 Log_Dir configuration element, 19-8
 LogPurgeDays configuration element, 19-8
 managing device types, 19-9
 manually configuring settings, 19-3
 MaxDeliverySleepSecs configuration element, 19-9
 MaxGlobalSleepSecs configuration element, 19-8
 MaxRequestSleepSecs configuration element, 19-8
 MaxRowsTimesColumns configuration element, 19-9
 MinDeliverySleepSecs configuration element, 19-9
 MinGlobalSleepSecs configuration element, 19-8
 MinRequestSleepSecs configuration element, 19-8
 number connection attempts, 19-8
 number delivery attempts, 19-8
 NumDeliveryRetries configuration element, 19-8
 NumGlobalReTries configuration element, 19-8
 NumRequestRetries configuration element, 19-8
 permission settings, 19-2
 Scheduler configuration settings, 19-8

 seeding server cache, 7-25
 sleep time, 19-8
 using to seed cache, 7-25
 viewing information about active sessions, 19-10
aggregate tables, using, 7-4
AllowedLanguages configuration element, 15-5
AllowedLocales configuration element, 15-5
AllowRememberPassword configuration element, B-2
alternating bars, changing color, 18-9
alternative methods for starting and stopping system components, 4-6
AlwaysKeepSessionAffiliation configuration parameter, 6-7
ampersand in time zone values, 14-3
analyses
 blocking, 18-13
 default currency, 16-1
 default header and footer, 18-18
 general configuration tasks, 18-2
 navigation and drill down, 18-3
 unattended time, B-2
 view defaults, 18-17
API, Oracle BI Systems Management, 22-9
application, sample, 1-10
archiving
 catalog folders, 17-31
 objects, 17-7
 Presentation Catalog, 17-31
attack on Web pages, B-2
attribute files, 17-3

B

backup and recovery, 25-1
bars, changing color for alternating bars, 18-9
BI domain, definition of, 1-4
BI Publisher, configuring toolbar, 18-12
biee_domain.xml, disabling Fusion Middleware Control from managing configuration settings, D-1
blocking analyses, 18-13
briefing books, table of contents, 18-13
BulkFetchBufferSize (bytes) configuration element, 19-6

C

- CACertificateDir configuration element, 19-7
- CACertificateFile configuration element, 19-7
- cache
 - about hits, 7-21
 - administering, 7-33
 - attributes for physical tables, 7-16
 - configuring event polling tables, 7-16
 - displaying global cache information, 7-26
 - effects of repository changes on, 7-20
 - enabling and disabling, 7-13
 - in a clustered environment, 7-11
 - managing settings, 7-33
 - monitoring and managing, 7-15
 - persistence time setting, 7-16
 - purging using Administration Tool, 7-27
 - purging using ODBC procedures, 7-16
 - purging when switching between repositories, 7-20
 - reasons for disabling, 7-15
 - seeding with a suite of queries, 7-24
 - seeding with agents, 7-25
 - setting max entry size, 7-13
 - setting max number of entries, 7-13
 - strategies, 7-21
 - using event polling tables, 7-28
 - using the Cache Manager, 7-25
 - using with row-level database security, 7-23
 - viewing information in Cluster Manager, 6-11
- Cache Manager
 - about and opening, 7-25
 - displaying global cache information, 7-26
 - purging cache, 7-27
 - using, 7-25
- capacity, managing, 5-1
- captions, localizing for Presentation Catalog, 17-30
- catalog folders
 - archiving, 17-31
 - unarchiving, 17-31
- Catalog Manager
 - about, 17-11
 - archiving a Presentation Catalog, 17-31
 - components of, 17-16
 - copying and pasting items between Presentation Catalogs, 17-19
 - creating reports to display Presentation Catalog data, 17-30
 - localizing Presentation Catalog captions, 17-30
 - localizing Presentation Catalog text strings, 15-7
 - opening a Presentation Catalog, 17-15
 - previewing objects, 17-25
 - recommendations for working with, 17-12
 - renaming items, 17-22
 - searching for and replacing Presentation Catalog text, 17-28
 - searching for Presentation Catalog items, 17-18
 - setting permissions of items, 17-23
 - starting, 17-13
 - unarchiving a Presentation Catalog, 17-31
 - viewing and editing objects in XML, 17-26
 - working with item properties, 17-22
 - workspace, 17-16
- catalogs
 - file system guidelines, 17-4
 - validating, 17-9
- centrally managing
 - Java components using WebLogic Server administration console, 2-7
 - Oracle Business Intelligence using Fusion Middleware Control, 2-2
- CertificateFileName configuration element, 19-7
- CertificateVerifyDepth configuration element, 19-7
- certification information, 1-11
- CertPrivateKeyFileName configuration element, 19-7
- change/delete permission, 19-2
- checkpoints for usage tracking, 9-6
- CipherList configuration element, 19-7
- clickjacking, B-2
- client sessions, expiration time, 7-5
- cluster
 - about components in, 6-2
 - configuring for, 6-6
 - managing with Fusion Middleware Control, 6-5
 - shared files and directories, 5-4
 - shared Oracle BI Presentation Catalog, 5-5
 - shared repository publishing directory, 5-4
 - troubleshooting, 6-13
 - viewing and managing using Cluster Manager, 6-9
- Cluster Controller, configuring primary and secondary, 6-4
- Cluster Manager
 - See also* Cluster Server
 - about, 6-8
 - accessing when components are stopped, 6-8
 - activating Oracle BI Scheduler instances, 6-8
 - managing clustered servers, 6-12
 - viewing and managing cluster information, 6-9
 - viewing cache information, 6-11
 - viewing session information, 6-11
- Cluster Server
 - See also* Cluster Manager
 - viewing cache information, 6-11
- ColocatedOracleMapViewContextPath configuration element, 20-2
- colors, changing alternating bar color, 18-9
- comments and status overrides, configuring, 1-11
- components
 - Catalog Manager, 17-16
 - Java and system, 1-6
- Compound Layout default views, 18-19
- concurrent connections, maximum number, 19-5
- configuration
 - alternate BI Publisher toolbar, 18-12
 - cache, accessing, 7-33
 - changes for Presentation Services, B-1
 - default language, specifying for login screen, 15-4
 - file location, 3-5
 - general tasks for analyses, 18-2

- graphs and gauges, 18-6
- instanceconfig.xml file, B-1
- JavaHost service, B-3
- map views, 20-1
- marketing content server, 11-3, 11-4
- Smart View download, 11-4
- tables and pivot tables, 7-6, 12-1
- time zones, 14-4
- URL generation and resource file location, 21-1
- configuration settings
 - disabling Fusion Middleware Control from managing, D-1
 - methods for updating, 3-1
 - updating using a text editor, 3-3
 - updating using Fusion Middleware Control, 3-2
 - updating using Java program with Oracle BI Systems Management API, 22-12
 - updating using MBean Browser, 3-4
 - updating using Oracle BI Systems Management API, 3-3, 22-9
 - updating using WLST with Oracle BI Systems Management API, 22-9
- ControllerPollSeconds configuration parameter, 6-6
- CookieDomain configuration element, B-2
- CookiePath configuration element, B-2
- cookies, B-2
- copying and pasting content between catalogs, 17-19
- copyright text on maps, 20-4
- CREATE TABLE statements, for event polling tables, 7-31
- creating
 - device types, 19-9
 - reports to display Presentation Catalog data, 17-30
- CubeMaxPopulatedCells configuration element, 18-4
- CubeMaxRecords configuration element, 18-4
- currency
 - customized subject areas, 16-1
 - default for analyses, 16-1
 - user-preferred options, 16-2

D

- dashboards
 - headers and footers, 18-19
 - links to page, 18-11
 - managing, 18-1
 - page defaults, 18-19
 - section defaults, 18-19
 - view defaults, 18-17
- data sources, setting up, 1-11
- DATA_STORAGE_PATHS configuration parameter, 7-14
- database, back-end
 - bulk fetch buffer size, 19-6
 - job script path, 19-5
 - maximum concurrent connections field, 19-5
 - purging, 19-6
 - timeout specification, 19-5

- tuning and indexing, 7-4
- databases supported, 1-11
- DDL, Dynamic Library section parameters, A-25
- default RPD, viewing, 10-2
- DefaultName configuration element, 18-10
- DefaultRowsDisplayed configuration element, 18-5
- DefaultScriptPath configuration element, 5-6, 19-6
- deleting device types, 19-9
- delivery devices, 19-9
- delivery profiles, 19-9
- descriptions for repository objects, externalizing for localization, 15-16
- diagnostics
 - introduction, 8-1
 - log configuration file format example for BI Server, 8-7
 - log configuration file format example for JavaHost Server, 8-7
 - log configuration file settings, 8-7
 - log configuration files location, 8-6
 - log files, 8-2
 - log files location, 8-5
- dictionary, metadata, 18-2
- disabling Fusion Middleware Control from managing configuration settings, D-1
- display names for repository objects, externalizing for localization, 15-16
- double column support, associating a descriptor ID column and descriptor column, 15-19
- downloading Smart View, 11-4
- drill down, adding support for, 18-3
- DSN, Presentation Services, B-2
- Dynamic Library section (NQSCONFIG.INI file) parameters, about, A-25

E

- ECID, viewing log information, 8-2
- editing device types, 19-9
- embedded fonts, 18-8
- embedded maps in pages, 20-4
- EmbedFonts configuration element, 18-7
- EnableBookmarkURL configuration element, 18-11
- EnablePromptedURL configuration element, 18-11
- event polling tables
 - about, 7-16, 7-28
 - column names, 7-29
 - configuring, 7-16
 - CREATE TABLE statements, 7-31
 - making active, 7-32
 - populating, 7-32
 - setting up, 7-29
 - troubleshooting, 7-33
- exchanging metadata, 1-11
- exporting Presentation Catalog text strings, 15-7
- exposing Presentation Catalog text strings, 15-11
- Externalize Strings utility, using, 15-16

F

- failover, configuring for, 6-6
- FlashCLSID configuration element, 18-7
- FlashCodeBase configuration element, 18-7
- folders, archiving and unarchiving, 17-31
- fonts, embedded, 18-8
- footer for analyses, 18-18
- frame busting, B-2
- framework, actions, 11-2
- Fusion Middleware Control
 - MBean Browser, 2-10
 - navigation tree, 2-7
 - starting and stopping system and Java components, 4-4
 - starting to centrally manage Oracle Business Intelligence, 2-2
 - tool to monitor and configure system components, 1-8
 - user interface labels mapped to configuration file elements, C-1
 - using to centrally manage Oracle Business Intelligence, 2-2
 - using to configure the connection to the marketing content server, 11-3
 - using to update configuration settings, 3-2
 - WebLogic Server, 2-1

G

- gauge views, configuring, 18-6
- General section parameters (NQSSConfig.INI file)
 - about, A-9
 - AGGREGATE_PREFIX, A-32
 - CASE_SENSITIVE_CHARACTER_COMPARISON, A-11
 - DATE_DISPLAY_FORMAT, A-13
 - DATE_TIME_DISPLAY_FORMAT, A-13
 - LOCALE, A-9
 - NULL_VALUES_SORT_FIRST, A-13
 - SORT_ORDER_LOCALE, A-10
 - SORT_ORDER_LOCALE on UNIX, A-11
 - SORT_TYPE, A-11
 - TIME_DISPLAY_FORMAT, A-13
 - UPPERCASE_USERNAME_FOR_INITBLOCK, A-15
 - VIRTUAL_TABLE_PAGE_SIZE, A-14
 - WORK_DIRECTORY_PATHS, A-13
- global cache
 - about, 7-11
 - configuring, 7-14, 7-15
 - displaying information in Cache Manager, 7-26
 - setting up a network share for, 5-5
- graph views
 - common settings, 18-4
 - configuring, 18-6
 - cube settings, 18-4
 - fonts, 18-8
- green bar styling, 18-9
- guidelines
 - for Catalog Manager, 17-12

- for catalogs, 17-4
- for object names, 17-2

H

- header for analyses, 18-18
- high availability
 - deploying Oracle Business Intelligence for, 6-1
 - deployment diagram, 6-2
 - eliminating single points of failure, 6-3
 - managing using Fusion Middleware Control, 6-5
- horizontal scaling
 - about, 5-2
 - performing steps for, 5-3

I

- InIFrameRenderingMode configuration element, B-2
- inline frames, B-2
- installing and upgrading, 1-11
- instanceconfig.xml file
 - for Oracle BI Scheduler, 5-6
 - for Presentation Services, B-1

J

- Java components
 - definition of, 1-4, 1-6
 - in a clustered environment, 6-2
 - listed, 1-6
 - managing using WebLogic server administration console, 1-9
 - monitoring and starting and stopping using Fusion Middleware Control, 2-5
- Java program, using with Oracle BI Systems Management API to update configuration settings, 22-12
- JavaHost
 - diagnostic log configuration file format example, 8-7
 - service, B-3
- JavaHostProxy configuration parameter, 6-7
- JavaHostReadLimitInKB configuration element, 18-5
- JavaScript files, 18-14
- JMX MBeans
 - container, 1-7
 - using the MBean Browser to manage, 2-9
- Job Manager Configuration dialog box
 - accessing, 19-4
 - agents tab fields, 19-8
 - role of, 19-4
 - Scheduler Tab fields, 19-5
- job scripts path, 19-5

K

- KeepErrorLogFiles configuration element, 19-9

L

languages, default language, specifying for login screen, 15-4

links to dashboard pages, 18-11

listen address, setting, 5-7

localization

of repository metadata, 15-16

setting the Japanese locale on AIX systems, 15-16
SORT_ORDER_LOCALE settings on UNIX systems, 15-13

location

BI configuration files, 3-5

Presentation Catalog, 17-1

lock files, 17-3

locking mechanism for multiple concurrent administrators, 2-1

log files

See also query log, administering

agent log directory, viewing entries, 8-21

configuration file location, 8-6

levels, 8-8

manually configuring rotation policy, 8-4

message categories, 8-8

messages in, 8-20

Presentation Services, 8-14

rotation, 8-3, 8-9

viewing in Fusion Middleware Control, 8-2

log viewer utility

about, 8-13

interpreting log records, 8-14

running, 8-13

Log_Dir configuration element, 19-8

LogAllSqlStmts configuration element, 19-6

logging levels

configuring using Fusion Middleware Control, 8-3

described, 8-12

enabling for individual users, 8-11

log viewer utility, 8-13, 8-14

manually configuring, 8-4

overriding with session variable, 8-11

user logging levels, 8-13

logical architecture for Oracle Business

Intelligence, 1-3

login screen, specifying default language, 15-4

LogPurgeDays configuration element, 19-8

lookup

creating logical lookup columns, 15-23

creating logical lookup tables, 15-20

designating a logical table as a lookup table, 15-22

designing translation lookup tables in a mls schema, 15-19

function syntax, 15-22

guidelines for scaling, 5-2

in a clustered environment, 6-2

scaling horizontally, 5-3

starting and stopping on Windows and UNIX, 4-11

manually starting the Oracle Business Intelligence system, 4-2

map views

configuring, 20-1

configuring manually, 20-2

configuring MapViewer, 20-1

embedded in pages, 20-4

hardware requirements, 13-3

inserting text, 20-4

system requirements, 13-1

translating layers, 13-8

mapping configuration file elements with

corresponding Fusion Middleware Control user interface labels, C-1

MapViewer configuration, 20-1

marketing content server

configuring the connection to, 11-3, 11-4

using Fusion Middleware Control to configure the connection to, 11-3

Master BI Server, viewing in Cluster Manager, 6-9

MAX_ROWS_PER_CACHE_ENTRY configuration parameter, 7-14

MaxAgeMinutes configuration element, 18-11

MaxCells configuration element, 18-5

MaxDeliverySleepSecs configuration element, 19-9

MaxDropDownValues configuration element, 18-10

MaxEntries configuration element, 7-34

MaxExecThreads configuration element, 19-6

MaxExpireMinutes configuration element, 7-34

MaxFollowLinks configuration element, 18-10

MaxGlobalSleepSecs configuration element, 19-8

MaxRecords configuration element, 20-2

MaxRequestSleepSecs configuration element, 19-8

MaxRowsTimesColumns configuration element, 19-9

MaxVisibleColumns configuration element, 18-5

MaxVisiblePages configuration element, 18-5

MaxVisibleRows configuration element, 18-5

MaxVisibleSections configuration element, 18-5

MBean Browser

updating configuration settings, 3-4

using to manage Oracle Business Intelligence, 2-9

MDX Member Name Cache section parameters (NQSConfig.INI file)

DATA_STORAGE_PATH, A-31

ENABLE, A-31

MAX_MEMBER_PER_LEVEL, A-31

MAX_SIZE_PER_USER, A-31

memory requirements, 1-11

messages in log files, 8-20

metadata

exchanging, 1-11

importing into IBM DB2 or Oracle Database, 1-11

metadata dictionary, 18-2

metrics, viewing, 7-2

M

Managed Server, 1-7

confirming whether running, 4-12

definition of, 1-4

- migrating to a new environment, 24-1
- migration, whole server, 24-1
- MinDeliverySleepSecs configuration element, 19-9
- MinExecThreads configuration element, 19-6
- MinExpireMinutes configuration element, 7-34
- MinGlobalSleepSecs configuration element, 19-8
- minimum disk space, 1-11
- MinRequestSleepSecs configuration element, 19-8
- MinUserExpireMinutes configuration element, 7-34
- modes for opening catalogs, 17-13
- monitoring system metrics, 7-1
- multilingual data support, 15-18
- multilingual support, lookup, 15-19
- multiple host logical architecture, 1-6

N

- navigation tree in Fusion Middleware Control, 2-7
- navigation, adding support for in analyses, 18-3
- node manager
 - daemon process, 1-4
 - starting on Windows and UNIX, 4-3
- nQLogViewer utility, interpreting log records, 8-14
- NQSCONFIG.INI file
 - comments, adding, A-2
 - configuration file parameter entries, rules for, A-1
 - configuring parameters for clustering, load balancing, and failover, 7-15
 - Dynamic Library section parameters, A-25
 - General section, about, A-9
 - Optimization Flags section parameters, A-31
 - parameters and syntax, about, A-1
 - parsing rules, A-1
 - Query Result Cache section, parameters, A-3
 - Repository section parameters, A-3
 - security section parameters, A-15
 - Server section parameters, A-17
 - Usage Tracking section parameters, A-26
- NumDBConnections configuration element, 19-5
- NumDeliveryRetries configuration element, 19-8
- NumGlobalRetries configuration element, 19-8
- NumRequestRetries configuration element, 19-8

O

- object names, guidelines for, 17-2
- objects
 - attribute files for, 17-3
 - committing to production, 17-7
 - guidelines for names, 17-2
 - lockfiles for, 17-3
 - previewing in Catalog Manager, 17-25
 - searching for, 17-18
 - viewing and editing in XML, 17-26
 - working with properties, 17-22
- ODBC procedures
 - about syntax for, 7-17
 - for Presentation Services shared request key, 7-18
 - for purging cache, 7-16

- for purging SAP/BW data sources, 7-19
- offline mode for opening catalogs, 17-13
- online mode for opening catalogs, 17-13
- opening catalogs, 17-13
- operating systems, supported, 1-11
- OPMN
 - about, 1-9
 - definition of, 1-4
 - using opmnctl command line to start and stop
 - System components, 4-6
- Optimization Flags section parameters (NQSCONFIG.INI file)
 - parameters, about, A-31
 - STRONG_DATETIME_TYPE_CHECKING, A-31
- options, delivery for agents, 19-9
- Oracle BI Catalog Manager
 - See Catalog Manager
- Oracle BI Dashboards
 - See dashboards
- Oracle BI Presentation Services Plug-in, configuring
 - for high availability, 6-7
- Oracle BI repository
 - disallowing updates, 7-5
 - localizing repository metadata, 15-16
 - metadata dictionary, 18-2
 - repository publishing directory, 5-4
 - restarting Oracle BI processes, 4-1
 - sharing, 10-3
 - specifying password for, 10-3
 - understanding version numbers, 10-2
 - uploading using Fusion Middleware Control, 10-2
 - uploading using System MBeans, 10-3
- Oracle BI Server
 - diagnostic log configuration file format
 - example, 8-7
 - loading repository, 10-2
 - viewing sessions in Cluster Manager, 6-11
 - Windows, starting or restarting, 4-8
- Oracle BI Systems Management API
 - accessing API methods, 22-2
 - JMX Admin MBeans and Methods, 22-25
 - purpose of, 22-1
 - using to manage and configure Oracle Business Intelligence, 22-2
 - using to programmatically start and stop Oracle Business Intelligence, 4-9
 - using to scale out for high availability and performance, 22-14
 - using to start and stop Oracle Business Intelligence, 22-7
 - using to update configuration settings, 3-3, 22-9
- Oracle Business Intelligence
 - deploying for high availability, 6-1
 - logical architecture explained, 1-3
 - scaling, 5-1
- Oracle Scorecard and Strategy Management,
 - configuring comments and status overrides for, 1-11
- Oracle WebLogic Server Admin Console, starting and

stopping Java components, 4-5

P

PassphraseFileName configuration element, 19-7

password

remembering, B-2

specifying for repository upon upload, 10-3

pasting items between Presentation Catalogs, 17-19

patching

about, 23-1

components, 23-1

determining current patch levels, 23-6

platform patches, 23-2

rolling back a patch, 23-4

PauseOnStartup configuration element, 19-6

PERF_PREFER_IN_LISTS database feature, 7-29

performance

about managing, 7-4

disallowing repository updates, 7-5

for Web client, 7-35

setting up static file caching, 7-35

tuning and indexing data sources, 7-4

using aggregate tables, 7-4

viewing metrics, 7-2

permissions

setting for items, 17-23

setting in Catalog Manager, 17-23

settings for agents, 19-2

pivot table views

common settings, 18-4

configuration, 7-6, 12-1

cube settings, 18-4

platforms, supported, 1-11

PoolTimeout (minutes) configuration element, 19-5

POPULATE_AGGREGATE_ROLLUP_HITS

configuration parameter, 7-14

ports

specifying, 5-7

viewing, 5-8

precedence order for time zones, 14-3

Presentation Catalog

about, 17-1

administering, 17-11, 17-22

archiving, 17-31

Catalog Manager workspace, about, 17-16

Catalog Manager, setting permissions

through, 17-23

Catalog Manager, starting, 17-13

committing an object to production, 17-7

configuring for a full text search, 17-32

copying and pasting objects between, 17-19

creating new, 17-7

creating reports to display data, 17-30

exporting text strings for, 15-7

exposing text strings for, 15-11

localizing captions for, 17-30

location of, 17-1

maintaining manually, 17-6

maintaining using Fusion Middleware

Control, 17-6

manually changing additional configuration

settings for, 17-6

manually creating a new one, 17-7

modes for opening in Catalog Manager, 17-13

opening, 17-13

recommendations for working with, 17-12

rename items, about using Catalog Manager

to, 17-22

replicating, 17-40

searching for and replacing text, 17-28

setting object permissions, 17-23

setting permissions of objects, 17-23

setting up a network share for, 5-5

specifying shared location for, 10-3

unarchiving, 17-31

viewing and editing objects in XML, 17-26

Presentation Services

cache, 7-33

configuring for high availability, 6-7

log message structure, 8-20

logging in, 8-14

query cache, 7-18

Replication Agent, 17-42

specifying default language, for login screen, 15-4

previewing objects from Catalog Manager, 17-25

production, committing objects to, 17-7

programmatically starting and stopping Oracle

Business Intelligence using the Oracle BI Systems

Management API, 4-9

properties of catalog objects, 17-22

publish agents for Subscription privilege, 19-2

PurgeInstDays configuration element, 19-6

PurgeIntervalMinutes configuration element, 19-6

purging cache

for SAP/BW data sources, 7-18

result records, 7-18

using Administration Tool, 7-27

using ODBC procedures, 7-16

when switching between repositories, 7-20

Q

query caching

about hits, 7-21

advantages of, 7-8

attributes for physical tables, 7-16

cache strategies, 7-21

configuring event polling tables, 7-16

cost of caching, about, 7-9

displaying global cache information, 7-26

effects of repository changes on, 7-20

enabling and disabling, 7-13

monitoring and managing, 7-15

Presentation Services query cache, 7-18

purging using Administration Tool, 7-27

purging using ODBC procedures, 7-16

purging when switching between

repositories, 7-20

reasons for disabling, 7-15

- refresh interval, setting for XML data sources, 7-10
- seeding with a suite of queries, 7-24
- seeding with agents, 7-25
- setting max entry size, 7-13
- setting max number of entries, 7-13
- using event polling tables, 7-28
- using the Cache Manager, 7-25
- using with row-level database security, 7-23

- query environment, administering
 - query logs, administering, 8-10
 - usage tracking, administering, 9-1

- query logs
 - administering, 8-10
 - configuring, 8-11
 - disabling user logging levels, 8-13
 - interpreting the log records, 8-14
 - log viewer utility, 8-13
 - logging levels, 8-12
 - setting logging levels, 8-11
 - setting user logging levels, 8-13

- Query Result Cache section parameters (NQSConfig.INI file), A-3
 - CACHE_POLL_SECONDS, A-9
 - CLUSTER_AWARE_CACHE_LOGGING, A-9
 - DATA_STORAGE_PATHS, A-4
 - ENABLE, A-4
 - GLOBAL_CACHE_STORAGE_PATH, A-8
 - MAX_CACHE_ENTRIES, A-6
 - MAX_CACHE_ENTRY_SIZE, A-6
 - MAX_GLOBAL_CACHE_ENTRIES, A-9
 - MAX_ROWS_PER_CACHE_ENTRY, A-5
 - MAX_SUBEXPR_SEARCH_DEPTH, A-8
 - POPULATE_AGGREGATE_ROLLUP_HITS, A-6
 - USE_ADVANCED_HIT_DETECTION, A-7

R

- recovery and backup, 25-1
- refresh interval, setting for XML data sources, 7-10
- relational data source adapters, 1-11
- RemoteOracleMapViewAbsoluteURL configuration element, 20-2
- renaming items
 - about, 17-22
 - with reference updates, 17-22
 - without reference updates, 17-22
- replacing Presentation Catalog text, 17-28
- replication
 - catalogs, 17-40
 - Oracle BI Presentation Services Replication Agent, 17-42
 - resuming, 17-49
 - sawrepaj utility, 17-42
- repository file, disallowing updates, 7-5
- repository publishing directory, setting up a network share for, 5-4
- Repository section parameters (NQSConfig.INI file), A-3
- repository variables, cache purging

- considerations, 7-20
- requirements, system, 1-11
- resource files, location, 21-1
- result records, for purging cache, 7-18
- resuming replication, 17-49
- rollover, described, 9-6
- row-level database security, ensuring correct cache results with, 7-23

S

- SA System subject area, delivery options, 19-9
- sample application, 1-10
- sawrepaj utility, 17-42
- scaling your system
 - about, 5-1
 - performing steps for, 5-3
 - recommendations, 5-2
 - using Fusion Middleware Control, 5-6
 - using the Oracle BI Systems Management API, 22-14
- Scheduler
 - activating inactive instances of, 6-8
 - computer running, 19-3
 - configuration options, 19-4
 - configuration tasks for, 1-11
 - configuring for high availability, 5-5
 - configuring primary and secondary, 6-4
 - configuring Scheduler Tab options, 19-5
 - DefaultScriptPath configuration element, 5-6
 - execution threads, 19-6
 - SchedulerScriptPath configuration element, 5-6
 - schema tables, 19-5
 - scripts, setting up a network share for, 5-5
 - SchedulerScriptPath configuration element, 5-6, 19-5
 - ScriptRPCPort configuration element, 19-7
 - searching for
 - and replacing catalog text, 17-28
 - catalog objects, 17-18
 - SectionSliderDefault configuration element, 18-7
 - SectionSliderLimit configuration element, 18-7
 - Security section parameters (NQSConfig.INI file)
 - DEFAULT_PRIVILEGES, A-15
 - parameters, about, A-15
 - PROJECT_INACCESSIBLE_COLUMN_AS_NULL, A-15
 - SSL, A-16
 - SSL_CA_CERTIFICATE_FILE, A-17
 - SSL_CERT_VERIFICATION_DEPTH, A-17
 - SSL_CERTIFICATE_FILE, A-16
 - SSL_CIPHER_LIST, A-17
 - SSL_PK_VERIFY_PEER, A-17
 - SSL_PRIVATE_KEY_FILE, A-16
 - SSL_TRUSTED_PEER_DNS, A-17
 - Security Sensitive option, for session variables, 7-24
 - Server section parameters (NQSConfig.INI file)
 - AUTOMATIC_RESTART, A-24
 - CLIENT_MGMT_THREADS_MAX, A-21
 - CLUSTER_PARTICIPANT, A-22
 - DB_GATEWAY_THREAD_RANGE, A-20

- DB_GATEWAY_THREAD_STACK_SIZE, A-20
- ENABLE_DB_HINTS, A-22
- INIT_BLOCK_CACHE_ENTRIES, A-21
- MAX_DRILLDOWN_INFO_CACHE_ENTRIES, A-20
- MAX_DRILLDOWN_QUERY_CACHE_ENTRIES, A-21
- MAX_EXPANDED_SUBQUERY_PREDICATES, A-20
- MAX_QUERY_PLAN_CACHE_ENTRIES, A-20
- MAX_REQUEST_PER_SESSION_LIMIT, A-19
- MAX_SESSION_LIMIT, A-18
- parameters, about, A-17
- PREVENT_DIVIDE_BY_ZERO, A-22
- REPOSITORY_PUBLISHING_DIRECTORY, A-23
- REQUIRE_PUBLISHING_DIRECTORY, A-24
- RPC_SERVICE_OR_PORT, A-21
- SERVER_THREAD_RANGE, A-19
- SERVER_THREAD_STACK_SIZE, A-19
- ServerPollSeconds configuration parameter, 6-6
- ServerPortNumber configuration element, 19-6
- service levels, monitoring, 7-1
- session variables
 - for time zones, 14-4
 - Security Sensitive option, 7-24
 - using to translate strings, 15-17
- sessions
 - expiration time, 7-5
 - viewing information about active agent, 19-10
- settings for time zones, 14-4
- shared files and directories, clustering requirements, 5-4
- Shared Location option
 - for catalogs, 10-3
 - for Oracle BI repository, 10-3
- single points of failure, eliminating, 6-3
- SIZE parameters (NQSSConfig.INI file)
 - USE_LONG_DAY_NAMES, A-15
 - USE_LONG_MONTH_NAMES, A-15
- Smart View download, 11-4
- specifications for time zones, 14-3
- SSL information, 1-11
- SSL Trusted Peer DN's configuration field, 19-7
- starting and stopping
 - Administration Server, 4-9
 - Catalog Manager, 17-13
 - Fusion Middleware Control, 2-2
 - Java components, 4-5
 - Managed Server, 4-11
 - Oracle BI components, 4-1
 - Oracle Business Intelligence system manually, 4-2
 - Oracle Business Intelligence using the Oracle BI Systems Management API, 22-7
 - system and Java components, 4-4
 - system components, 4-6
 - system components with a Windows service, 4-8
 - system components with opmnctl command line, 4-6
- static file caching, setting up, 7-35
- status
 - displaying using the Oracle BI Systems Administration API, 22-2
 - viewing for components, in Fusion Middleware Control, 2-4
- STORAGE_DIRECTORY parameter
 - user tracking log files, selecting an output location for, 9-3
- strings
 - externalizing for localization, 15-16
 - translating using session variables, 15-17
- subject area, SA System, 19-9
- supported installation types, 1-11
- SyndicatedOracleMapViewContextPath
 - configuration element, 20-2
- system administration
 - introduction and overview, 1-1
 - links to typical tasks, 1-2
 - tasks, 1-1
 - tools for managing Oracle Business Intelligence, 1-8
- system components
 - definition of, 1-4, 1-7
 - displaying names, 4-7
 - in a clustered environment, 6-2
 - listed, 1-6
 - managing using Fusion Middleware Control, 2-4
 - scaling, 5-2
 - scaling vertically and horizontally, 5-6
 - specifying ports for, 5-7
 - starting and stopping, 5-8
 - viewing host, instance, and port information for, 5-8
 - viewing status for, 5-8
- system metrics, viewing, 7-2
- system requirements, 1-11
- system startup, troubleshooting, 4-13

T

- table of contents for briefing books, 18-13
- table views
 - common settings, 18-4
 - configuration, 7-6, 12-1
 - configuring maximum number of rows, 7-6
- templates
 - write back, 18-22
 - write back example, 18-24
- temporary files, location of, 19-6
- TemporaryFilePath configuration element, 19-6
- text editor to manually update configuration settings, 3-3
- text strings
 - exporting for Presentation Catalog, 15-7
 - exposing for Presentation Catalog, 15-11
 - localizing for Presentation Catalog, 15-7, 17-30
- third-party tools, 1-11
- time zones
 - precedence order, 14-3
 - session variables, 14-4

- setting, 14-2
- setting for users, 14-1
- settings, 14-4
- specifications, 14-3
- specifying values, 14-3
- usage, 14-1
- user-preferred, 14-3
- toolbar for BI Publisher, 18-12
- topics of interest in other guides, 1-10
- troubleshooting
 - clustered environments, 6-13
 - event polling tables, 7-33
 - system startup, 4-13

U

- UI hints, propagating for ADF data sources, 1-11
- UnaccessedRunningTimeoutMinutes configuration element, B-2
- unarchiving
 - catalog folders, 17-31
 - catalogs, 17-31
 - objects, 17-7
- UNIX
 - Japanese locale on AIX, 15-16
 - SORT_ORDER_LOCALE parameter, 15-13, A-11
- updating configuration settings using Oracle BI Systems Management API, 22-9
- URLs, generation and resource file location, 21-1
- usage tracking
 - administering, 9-1
 - error accessing usage tracking output file, 9-4
 - file naming conventions, about and example, 9-4
 - output file, 9-4
 - output file column behavior, 9-5
 - output file format, 9-4
 - output location, selecting, 9-3
 - performance considerations, 9-6
- usage tracking log files, usage tracking data (table), 9-6
- Usage Tracking section parameters (NQSConfig.INI file)
 - BUFFER_SIZE, A-30
 - BUFFER_TIME_LIMIT_SECONDS, A-30
 - CHECKPOINT_INTERVAL_MINUTES, A-28
 - CODE_PAGE, A-29
 - CONNECTION_POOL, A-30
 - DIRECT_INSERT, A-27
 - ENABLE, A-27
 - FILE_ROLLOVER_INTERVAL_MINUTES, A-28
 - MAX_INSERTS_PER_TRANSACTION, A-30
 - NUM_INSERT_THREADS, A-30
 - parameters, about, A-26
 - PHYSICAL_TABLE_NAME, A-29
 - STORAGE_DIRECTORY, A-28
- user interface, setting the default language, 15-4
- users
 - preferred time zone, 14-3
 - session log-off period, 7-5
 - time zones, 14-1

- utilities
 - log viewer utility, 8-13, 8-14
 - sawrepaj, 17-42

V

- validating catalogs, 17-9
- validation helper functions, 18-16
- version numbers, for Oracle BI repository, 10-2
- vertical scaling
 - about, 5-2
 - using Fusion Middleware Control, 5-6
- view defaults, 18-17
- viewing device types, 19-9
- views, default in Compound Layout, 18-19

W

- Web client, improving performance for, 7-35
- WebLogic scripting tool for managing domains, 1-10
- WebLogic server administration console
 - centrally managing Java components, 2-7
 - tool to manage Java components, 1-9
- whole server migration, 24-1
- Windows, service for starting and stopping system components, 4-8
- WLST
 - tool to manage WebLogic server domains, 1-10
 - using with Oracle BI Systems Management API to update configuration settings, 22-9
- workspace, Catalog Manager, 17-12, 17-16, 17-17
- write back
 - limitations, 18-20
 - table, 18-21
 - template, 18-22
 - template example, 18-24

X

- XML data sources, setting refresh interval, 7-10
- XML message files, 18-17
- XML message files, view defaults, 18-17

Z

- zones, time, 14-1