

Oracle® Fusion Middleware

Administering Oracle Fusion Middleware

12c (12.1.2)

E36206-04

December 2014

Describes how to manage Oracle Fusion Middleware, including how to start and stop Oracle Fusion Middleware, how to configure and monitor components, how to back up and recover your environment and how to move from a source environment to a target environment.

Oracle Fusion Middleware Administering Oracle Fusion Middleware, 12c (12.1.2)

E36206-04

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved.

Primary Author: Helen Grembowicz

Contributing Author: Vinaye Misra

Contributors: Mike Blevins, Nick Fry, Greg Cook, Harry Hsu, Christine Jacobs, Srini Indla, Pavana Jain, Rama Kalava, Gopal Kirsur, Kenneth Ma, Dan MacKinnon, Manoj Nayak, Mark Nelson, Praveen Sampath, Sunita Sharma

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, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	xxv
Audience	xxv
Documentation Accessibility	xxv
Related Documents	xxv
Conventions	xxvi
What's New in This Guide?	xxvii
Updates for Release 12c Documentation Refresh, December 2014	xxvii
New and Changed Features for Release 12c	xxvii
Part I Understanding Oracle Fusion Middleware	
1 Introduction to Oracle Fusion Middleware	
1.1 What Is Oracle Fusion Middleware?	1-1
1.2 Oracle Fusion Middleware Components	1-1
Part II Basic Administration	
2 Getting Started Managing Oracle Fusion Middleware	
2.1 Overview of Oracle Fusion Middleware Administration Tools	2-1
2.2 Getting Started Using Oracle Enterprise Manager Fusion Middleware Control	2-3
2.2.1 Displaying Fusion Middleware Control	2-3
2.2.2 Using Fusion Middleware Control Help	2-5
2.2.3 Navigating Within Fusion Middleware Control	2-5
2.2.4 Understanding Users and Roles for Fusion Middleware Control	2-8
2.2.5 Viewing and Managing the WebLogic Domain	2-8
2.2.6 Viewing and Managing Components	2-9
2.3 Getting Started Using Oracle WebLogic Server Administration Console	2-9
2.3.1 Displaying the Oracle WebLogic Server Administration Console	2-10
2.3.2 Locking the WebLogic Server Configuration	2-11
2.4 Getting Started Using the Oracle WebLogic Scripting Tool (WLST)	2-12
2.4.1 Using WLST with Java Components	2-12
2.4.2 Using Custom WLST Commands	2-13
2.4.3 Using WLST Commands with System Components	2-13

2.5	Getting Started Using the Fusion Middleware Control MBean Browsers	2-14
2.5.1	Using the System MBean Browser	2-15
2.5.2	Using the MBeans for a Selected Application	2-16
2.6	Managing Components.....	2-16
2.7	Changing the Administrative User Password.....	2-16
2.7.1	Changing the Administrative User Password Using the Command Line.....	2-17
2.7.2	Changing the Administrative User Password Using the Administration Console	2-17
2.8	Configuring Node Manager	2-17
2.8.1	Configuring Node Manager to Start Managed Servers	2-17
2.8.2	Configuring Node Manager to Use the OPSS Keystore Service.....	2-18
2.9	Basic Tasks for Configuring and Managing Oracle Fusion Middleware	2-19

3 Cross-Component Wiring

3.1	Understanding Service Tables	3-1
3.2	Viewing Service Tables	3-1
3.3	Cross-Component Wiring.....	3-2
3.3.1	Wiring Oracle HTTP Server to the Administration Server.....	3-2
3.3.2	Routing Applications Through Oracle HTTP Server to Oracle WebLogic Server.....	3-4

4 Starting and Stopping Oracle Fusion Middleware

4.1	Overview of Starting and Stopping Procedures.....	4-1
4.2	Starting and Stopping Oracle WebLogic Server Administration and Managed Servers.	4-1
4.2.1	Starting and Stopping Administration Server.....	4-2
4.2.2	Starting and Stopping Node Manager.....	4-2
4.2.3	Starting and Stopping Managed Servers.....	4-2
4.2.3.1	Starting and Stopping Managed Servers Using Fusion Middleware Control.....	4-2
4.2.3.2	Starting and Stopping Managed Servers Using Scripts	4-3
4.2.4	Enabling Servers to Start Without Supplying Credentials	4-3
4.2.5	Setting Up Oracle WebLogic Server as a Windows Service	4-4
4.3	Starting and Stopping Components.....	4-4
4.3.1	Starting and Stopping Components Using Fusion Middleware Control.....	4-4
4.3.2	Starting and Stopping Components Using the Command Line.....	4-5
4.3.2.1	Starting and Stopping Java Components.....	4-5
4.3.2.2	Starting and Stopping System Components.....	4-5
4.4	Starting and Stopping Fusion Middleware Control	4-6
4.5	Starting and Stopping Applications.....	4-6
4.5.1	Starting and Stopping Java EE Applications Using Fusion Middleware Control	4-7
4.5.2	Starting and Stopping Java EE Applications Using WLST.....	4-7
4.6	Starting and Stopping Your Oracle Fusion Middleware Environment	4-7
4.6.1	Starting an Oracle Fusion Middleware Environment	4-7
4.6.2	Stopping an Oracle Fusion Middleware Environment	4-8
4.7	Starting and Stopping: Special Topics	4-9
4.7.1	Starting and Stopping in High Availability Environments.....	4-9
4.7.2	Forcing a Shutdown of Oracle Database	4-9

5 Managing Ports

5.1	About Managing Ports	5-1
5.2	Viewing Port Numbers	5-1
5.2.1	Viewing Port Numbers Using the Command Line	5-1
5.2.2	Viewing Port Numbers Using Fusion Middleware Control	5-2
5.3	Changing the Port Numbers Used by Oracle Fusion Middleware	5-2
5.3.1	Changing the Oracle WebLogic Server Listen Ports	5-3
5.3.1.1	Changing the Oracle WebLogic Server Listen Ports Using the Administration# Console.....	5-3
5.3.1.2	Changing the Oracle WebLogic Server Listen Ports Using WLST.....	5-3
5.3.2	Changing the Oracle HTTP Server Listen Ports.....	5-3
5.3.2.1	Enabling Oracle HTTP Server to Run as Root for Ports Set to Less Than 1024# (UNIX Only)	5-4
5.3.2.2	Changing the Oracle HTTP Server Non-SSL Listen Port in a WebLogic Server# Domain.....	5-4
5.3.2.3	Changing the Oracle HTTP Server SSL Listen Port in a WebLogic Server # Domain.....	5-4
5.3.2.4	Changing the Oracle HTTP Server Listen Ports in a Standalone Domain.....	5-5
5.3.3	Changing the Oracle Database Net Listener Port.....	5-5
5.3.3.1	Changing the KEY Value for an IPC Listener	5-7

Part III Secure Sockets Layer

6 Configuring SSL in Oracle Fusion Middleware

6.1	How SSL Works	6-1
6.1.1	What SSL Provides	6-1
6.1.2	About Private and Public Key Cryptography	6-2
6.1.3	Keystores and Wallets.....	6-3
6.1.4	How SSL Sessions Are Conducted.....	6-3
6.2	About SSL in Oracle Fusion Middleware.....	6-5
6.2.1	SSL in the Oracle Fusion Middleware Architecture	6-5
6.2.2	Keystores and Oracle Wallets	6-6
6.2.3	Authentication Modes.....	6-7
6.2.4	Tools for SSL Configuration.....	6-7
6.3	Configuring SSL for Configuration Tools	6-8
6.3.1	Oracle Enterprise Manager Fusion Middleware Control	6-8
6.3.2	Oracle WebLogic Server Administration Console.....	6-8
6.3.3	WLST Command-Line Tool	6-8
6.4	Configuring SSL for the Web Tier	6-8
6.4.1	Configuring Load Balancers.....	6-9
6.4.2	Enabling SSL for Oracle HTTP Server Virtual Hosts	6-9
6.4.2.1	Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using# Fusion Middleware Control.....	6-9
6.4.2.2	Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using# WLST	6-11
6.4.2.3	Enable SSL for Outbound Requests from Oracle HTTP Server.....	6-12
6.5	Configuring SSL for the Middle Tier	6-13

6.5.1	Configuring SSL for Oracle WebLogic Server	6-13
6.5.1.1	Inbound SSL to Oracle WebLogic Server	6-13
6.5.1.2	Outbound SSL from Oracle WebLogic Server.....	6-13
6.5.1.2.1	Outbound SSL from Oracle Platform Security Services to LDAP.....	6-13
6.5.1.2.2	Outbound SSL from Oracle Platform Security Services to Oracle # Database.....	6-14
6.5.1.2.3	Outbound SSL from LDAP Authenticator to LDAP.....	6-14
6.5.1.2.4	Outbound SSL to Database	6-14
6.5.2	Client-Side SSL for Applications	6-14
6.6	Configuring SSL for the Data Tier	6-14
6.6.1	Configuring SSL for the Database	6-15
6.6.1.1	SSL-Enable Oracle Database	6-15
6.6.1.2	SSL-Enable a Data Source	6-17
6.7	Advanced SSL Scenarios.....	6-18
6.7.1	Hardware Security Modules and Accelerators	6-18
6.7.2	CRL Integration with SSL.....	6-19
6.7.2.1	Configuring CRL Validation for a Component.....	6-19
6.7.2.2	Manage CRLs on the File System.....	6-20
6.7.2.3	Test a Component Configured for CRL Validation.....	6-21
6.7.3	Oracle Fusion Middleware FIPS 140-2 Settings	6-21
6.7.3.1	FIPS-Configurable Products	6-21
6.7.3.2	Setting the SSLFIPS_140 Parameter	6-21
6.7.3.3	Selecting Cipher Suites	6-22
6.7.3.4	Other Configuration Parameters.....	6-22
6.8	Best Practices for SSL	6-22
6.8.1	Best Practices for Administrators	6-22
6.8.2	Best Practices for Application Developers	6-23
6.9	WLST Reference for SSL	6-23
6.9.1	addCertificateRequest.....	6-24
6.9.1.1	Description	6-24
6.9.1.2	Syntax	6-24
6.9.1.3	Example.....	6-25
6.9.2	addSelfSignedCertificate	6-25
6.9.2.1	Description	6-25
6.9.2.2	Syntax	6-25
6.9.2.3	Example.....	6-25
6.9.3	changeWalletPassword	6-25
6.9.3.1	Description	6-26
6.9.3.2	Syntax	6-26
6.9.3.3	Example.....	6-26
6.9.4	configureSSL	6-26
6.9.4.1	Description	6-26
6.9.4.2	Syntax	6-26
6.9.4.3	Examples.....	6-27
6.9.5	createWallet	6-27
6.9.5.1	Description	6-27
6.9.5.2	Syntax	6-27
6.9.5.3	Examples.....	6-27

6.9.6	deleteWallet	6-27
6.9.6.1	Description	6-27
6.9.6.2	Syntax	6-28
6.9.6.3	Example.....	6-28
6.9.7	exportWallet	6-28
6.9.7.1	Description	6-28
6.9.7.2	Syntax	6-28
6.9.7.3	Examples.....	6-28
6.9.8	exportWalletObject	6-29
6.9.8.1	Description	6-29
6.9.8.2	Syntax	6-29
6.9.8.3	Examples.....	6-29
6.9.9	getSSL	6-30
6.9.9.1	Description	6-30
6.9.9.2	Syntax	6-30
6.9.9.3	Example.....	6-30
6.9.10	getWalletObject	6-30
6.9.10.1	Description	6-30
6.9.10.2	Syntax	6-30
6.9.10.3	Examples.....	6-31
6.9.11	importWallet.....	6-31
6.9.11.1	Description	6-31
6.9.11.2	Syntax	6-31
6.9.11.3	Examples.....	6-31
6.9.12	importWalletObject	6-32
6.9.12.1	Description	6-32
6.9.12.2	Syntax	6-32
6.9.12.3	Examples.....	6-32
6.9.13	listWalletObjects	6-33
6.9.13.1	Description	6-33
6.9.13.2	Syntax	6-33
6.9.13.3	Examples.....	6-33
6.9.14	listWallets.....	6-33
6.9.14.1	Description	6-33
6.9.14.2	Syntax	6-33
6.9.14.3	Example.....	6-34
6.9.15	removeWalletObject	6-34
6.9.15.1	Description	6-34
6.9.15.2	Syntax	6-34
6.9.15.3	Examples.....	6-34
6.9.16	Properties Files for SSL	6-35
6.9.16.1	Structure of Properties Files.....	6-35
6.9.16.2	Examples of Properties Files.....	6-36

7 Managing Keystores, Wallets, and Certificates

7.1	Key and Certificate Storage in Oracle Fusion Middleware	7-1
7.1.1	Types of Keystores.....	7-1

7.1.1.1	About Oracle Wallet.....	7-1
7.1.1.2	About the JKS Keystore	7-2
7.1.1.3	About the Keystore Service (KSS) Keystore	7-2
7.1.2	Keystore Management Tools.....	7-2
7.2	Command-Line Interface for Keystores and Wallets	7-3
7.3	Keystore Management	7-3
7.4	Wallet Management.....	7-3
7.4.1	About Wallets and Certificates	7-4
7.4.1.1	Password-Protected and Autologin Wallets	7-4
7.4.1.2	Self-Signed and Third-Party Wallets	7-4
7.4.1.3	Sharing Wallets Across Instances.....	7-5
7.4.1.4	Wallet Naming Conventions	7-5
7.4.2	Accessing the Wallet Management Page in Fusion Middleware Control.....	7-5
7.4.3	Managing the Wallet Life Cycle	7-6
7.4.4	Common Wallet Operations	7-6
7.4.4.1	Creating a Wallet Using Fusion Middleware Control	7-7
7.4.4.2	Creating a Wallet Using WLST.....	7-8
7.4.4.3	Creating a Self-Signed Wallet Using Fusion Middleware Control	7-8
7.4.4.4	Creating a Self-Signed Wallet Using WLST.....	7-9
7.4.4.5	Changing a Self-Signed Wallet to a Third-Party Wallet Using Fusion # Middleware Control.....	7-10
7.4.4.6	Changing a Self-Signed Wallet to a Third-Party Wallet Using WLST.....	7-10
7.4.4.7	Exporting a Wallet Using Fusion Middleware Control.....	7-10
7.4.4.8	Exporting a Wallet Using WLST	7-10
7.4.4.9	Importing a Wallet Using Fusion Middleware Control.....	7-11
7.4.4.10	Importing a Wallet Using WLST	7-11
7.4.4.11	Deleting a Wallet Using Fusion Middleware Control.....	7-11
7.4.4.12	Deleting a Wallet Using WLST.....	7-12
7.4.5	Managing the Certificate Life Cycle.....	7-12
7.4.6	Accessing the Certificate Management Page for Wallets in Fusion Middleware # Control.....	7-12
7.4.7	Common Certificate Operations.....	7-13
7.4.7.1	Adding a Certificate Request Using Fusion Middleware Control	7-13
7.4.7.2	Adding a Certificate Request Using WLST	7-14
7.4.7.3	Exporting a Certificate, Certificate Request, or a Trusted Certificate Using # Fusion Middleware Control.....	7-14
7.4.7.4	Exporting a Certificate, Certificate Request, or a Trusted Certificate Using # WLST	7-15
7.4.7.5	Importing a Certificate or a Trusted Certificate Using Fusion Middleware # Control	7-15
7.4.7.6	Importing a Certificate or a Trusted Certificate Using WLST	7-16
7.4.7.7	Deleting a Certificate Request, a Certificate, or a Trusted Certificate Using # Fusion Middleware Control.....	7-16
7.4.7.8	Deleting a Certificate Request, a Certificate, or a Trusted Certificate Using # WLST	7-17
7.4.7.9	Converting a Self-Signed Certificate into a Third-Party Certificate Using # Fusion Middleware Control.....	7-17
7.4.7.10	Converting a Self-Signed Certificate into a Third-Party Certificate Using # WLST	7-18

7.4.8	Wallet and Certificate Maintenance	7-19
7.4.8.1	Location of Wallets	7-19
7.4.8.2	Effect of Host Name Change on a Wallet	7-20
7.4.8.3	Changing a Self-Signed Wallet to a Third-Party Wallet	7-21
7.4.8.4	Replacing an Expiring Certificate in a Wallet	7-21

Part IV Deploying Applications

8 Understanding the Deployment Process

8.1	What Is a Deployer?.....	8-1
8.2	General Procedures for Moving from Application Design to Production Deployment ..	8-1
8.2.1	Designing and Developing an Application	8-1
8.2.2	Deploying an Application to Managed Servers	8-2
8.2.3	Automating the Migration of an Application to Other Environments.....	8-3
8.3	Diagnosing Typical Problems	8-4

9 Deploying Applications

9.1	Overview of Deploying Applications	9-1
9.1.1	What Types of Applications Can You Deploy?.....	9-1
9.1.2	Understanding Deployment, Redeployment, and Undeployment.....	9-2
9.2	Understanding and Managing Data Sources.....	9-3
9.2.1	Understanding Data Sources	9-3
9.2.2	Creating and Managing JDBC Data Sources	9-4
9.2.2.1	Creating a JDBC Data Source Using Fusion Middleware Control.....	9-5
9.2.2.2	Editing a JDBC Data Source Using Fusion Middleware Control	9-5
9.2.2.3	Monitoring a JDBC Data Source Using Fusion Middleware Control	9-6
9.2.2.4	Controlling a JDBC Data Source Using Fusion Middleware Control.....	9-6
9.2.2.5	Creating a GridLink Data Source Using Fusion Middleware Control	9-6
9.3	Deploying, Undeploying, and Redeploying Java EE Applications.....	9-7
9.3.1	Deploying Java EE Applications	9-7
9.3.1.1	Deploying Java EE Applications Using Fusion Middleware Control	9-7
9.3.1.2	Deploying Java EE Applications Using WLST.....	9-10
9.3.2	Undeploying Java EE Applications.....	9-11
9.3.2.1	Undeploying Java EE Applications Using Fusion Middleware Control.....	9-11
9.3.2.2	Undeploying Java EE Applications Using WLST.....	9-11
9.3.3	Redeploying Java EE Applications	9-11
9.3.3.1	Redeploying Java EE Applications Using Fusion Middleware Control	9-12
9.3.3.2	Redeploying Java EE Applications Using WLST.....	9-13
9.4	Deploying, Undeploying, and Redeploying Oracle ADF Applications	9-13
9.4.1	Deploying Oracle ADF Applications	9-14
9.4.1.1	Deploying ADF Applications Using Fusion Middleware Control	9-14
9.4.1.2	Deploying ADF Applications Using WLST or the Administration Console....	9-17
9.4.2	Undeploying Oracle ADF Applications	9-18
9.4.3	Redeploying Oracle ADF Applications.....	9-18
9.5	Managing Deployment Plans.....	9-20
9.6	About the Common Deployment Tasks in Fusion Middleware Control	9-20

9.7	Changing MDS Configuration Attributes for Deployed Applications	9-22
9.7.1	Changing the MDS Configuration Attributes Using Fusion Middleware Control	9-23
9.7.2	Changing the MDS Configuration Using WLST	9-25
9.7.3	Restoring the Original MDS Configuration for an Application	9-26

Part V Monitoring Oracle Fusion Middleware

10 Monitoring Oracle Fusion Middleware

10.1	Monitoring the Status of Oracle Fusion Middleware	10-1
10.1.1	Monitoring an Oracle WebLogic Server Domain	10-2
10.1.2	Monitoring an Oracle WebLogic Server Administration or Managed Server	10-3
10.1.3	Monitoring a Cluster	10-4
10.1.4	Monitoring a System Component	10-5
10.1.5	Monitoring Java EE Applications	10-5
10.1.6	Monitoring ADF Applications	10-6
10.1.7	Monitoring Applications Deployed to a Cluster	10-7
10.2	Viewing the Performance of Oracle Fusion Middleware	10-8
10.3	Viewing the Routing Topology	10-9

11 Managing Log Files and Diagnostic Data

11.1	Overview of Oracle Fusion Middleware Logging	11-1
11.1.1	Understanding Oracle Fusion Middleware HTTP Access Logging	11-1
11.1.2	Understanding Oracle Fusion Middleware Diagnostic Logging	11-2
11.2	Understanding ODL Messages and ODL Log Files	11-2
11.3	Viewing and Searching Log Files	11-5
11.3.1	Viewing Log Files and Their Messages	11-5
11.3.1.1	Viewing Log Files and Their Messages Using Fusion Middleware Control	11-5
11.3.1.2	Viewing Log Files and Their Messages Using WLST	11-7
11.3.2	Searching Log Files	11-9
11.3.2.1	Searching Log Files Using Fusion Middleware Control	11-9
11.3.2.1.1	Searching Log Files: Basic Searches	11-9
11.3.2.1.2	Searching Log Files: Advanced Searches	11-10
11.3.2.2	Searching Log Files Using WLST	11-11
11.3.3	Downloading Log Files	11-12
11.3.3.1	Downloading Log Files Using Fusion Middleware Control	11-12
11.3.3.2	Downloading Log Files Using WLST	11-13
11.4	Configuring Settings for Log Files	11-13
11.4.1	Changing Log File Locations	11-14
11.4.1.1	Changing Log File Locations Using Fusion Middleware Control	11-15
11.4.1.2	Changing Log File Locations Using WLST	11-15
11.4.2	Configuring Log File Rotation	11-15
11.4.2.1	Specifying Log File Rotation Using Fusion Middleware Control	11-16
11.4.2.2	Specifying Log File Rotation Using WLST	11-17
11.4.3	Setting the Level of Information Written to Log Files	11-17
11.4.3.1	Configuring Message Levels Using Fusion Middleware Control	11-19
11.4.3.2	Configuring Message Levels Using WLST	11-20

11.4.4	Specifying the Log File Format	11-21
11.4.4.1	Specifying the Log File Format Using Fusion Middleware Control	11-21
11.4.4.2	Specifying the Log File Format Using WLST	11-21
11.4.5	Specifying the Log File Locale	11-21
11.4.5.1	Specifying the Log File Encoding Using WLST	11-22
11.4.5.2	Specifying the Log File Encoding in logging.xml	11-22
11.5	Correlating Messages Across Log Files and Components	11-22
11.6	Configuring Tracing	11-24
11.6.1	Configuring and Using QuickTrace	11-24
11.6.1.1	Configuring QuickTrace Using Fusion Middleware Control	11-25
11.6.1.1.1	Configuring QuickTrace Using Fusion Middleware Control	11-25
11.6.1.1.2	Writing the Trace Messages to a File Using Fusion Middleware Control	11-26
11.6.1.2	Configuring QuickTrace Using WLST	11-26
11.6.1.2.1	Configuring QuickTrace Using WLST	11-27
11.6.1.2.2	Writing the Trace Messages to a File Using WLST	11-28
11.6.1.2.3	Disabling QuickTrace Using WLST	11-28
11.6.2	Configuring and Using Selective Tracing	11-29
11.6.2.1	Configuring Selective Tracing Using Fusion Middleware Control	11-29
11.6.2.1.1	Configuring Selective Tracing Using Fusion Middleware Control	11-29
11.6.2.1.2	Viewing Selective Traces Using Fusion Middleware Control	11-31
11.6.2.1.3	Disabling Selective Tracing Using Fusion Middleware Control	11-31
11.6.2.2	Configuring Selective Tracing Using WLST	11-31
11.6.2.2.1	Configuring Selective Tracing Using WLST	11-32
11.6.2.2.2	Viewing Selective Traces Using WLST	11-33
11.6.2.2.3	Disabling Selective Traces Using WLST	11-33

12 Diagnosing Problems

12.1	Understanding the Diagnostic Framework	12-1
12.1.1	About Incidents and Problems	12-3
12.1.1.1	Incident Flood Control	12-3
12.1.2	Diagnostic Framework Components	12-3
12.1.2.1	Automatic Diagnostic Repository	12-4
12.1.2.2	Diagnostic Dumps	12-6
12.1.2.3	Management MBeans	12-6
12.1.2.4	WLST Commands for Diagnostic Framework	12-6
12.1.2.5	ADRCI Command-Line Utility	12-6
12.2	How the Diagnostic Framework Works	12-7
12.3	Configuring the Diagnostic Framework	12-9
12.3.1	Configuring Diagnostic Framework Settings	12-10
12.3.2	Configuring Custom Diagnostic Rules	12-12
12.3.3	Configuring Problem Suppression	12-16
12.3.4	Configuring WLDF Watch and Notification for the Diagnostic Framework	12-18
12.4	Investigating, Reporting, and Solving a Problem	12-20
12.4.1	Roadmap—Investigating, Reporting, and Resolving a Problem	12-20
12.4.2	Viewing Problems and Incidents	12-22
12.4.2.1	Viewing Problems	12-22
12.4.2.2	Viewing Incidents	12-23

12.4.2.3	Querying Incidents.....	12-23
12.4.3	Analyzing Specific Problem Keys	12-24
12.4.4	Working with Diagnostic Dumps	12-25
12.4.4.1	Listing Diagnostic Dumps.....	12-26
12.4.4.2	Viewing a Description of a Diagnostic Dump.....	12-27
12.4.4.3	Executing Dumps	12-27
12.4.5	Configuring and Using Diagnostic Dump Sampling	12-27
12.4.5.1	Understanding Diagnostic Dump Sampling.....	12-27
12.4.5.2	Configuring Dump Sampling	12-29
12.4.5.2.1	Activating the Default Samples.....	12-29
12.4.5.2.2	Creating Dump Samplings	12-29
12.4.5.2.3	Modifying Dump Sampling Settings.....	12-30
12.4.5.2.4	Removing Dump Samplings.....	12-30
12.4.5.2.5	Enabling or Disabling All Dump Sampling	12-30
12.4.5.3	Listing Dump Samplings.....	12-31
12.4.5.4	Retrieving the Dump Sampling Output.....	12-31
12.4.5.4.1	Retrieving Dump Samples Using the executeDump Command	12-32
12.4.5.4.2	Retrieving Dump Samples Using the getSamplingArchives Command	12-32
12.4.6	Managing Incidents.....	12-32
12.4.6.1	Creating an Incident Manually.....	12-33
12.4.6.2	Creating an Aggregated Incident.....	12-34
12.4.6.3	Packaging an Incident.....	12-35
12.4.6.4	Purging Incidents	12-37
12.4.7	Generating an RDA Report	12-37
12.5	Managing and Running the Health Test Framework.....	12-38
12.5.1	Understanding the Health Test Framework.....	12-38
12.5.2	Understanding the Health Test Framework File Repository	12-39
12.5.3	Using the Health Test Framework Command Line.....	12-39
12.5.3.1	dfwhealthtestadminctl.sh Command Line	12-39
12.5.3.1.1	help	12-40
12.5.3.1.2	register	12-40
12.5.3.1.3	index	12-40
12.5.3.2	dfwhealthtestctl.sh Command Line.....	12-41
12.5.3.2.1	descstest.....	12-41
12.5.3.2.2	help	12-41
12.5.3.2.3	listrun	12-42
12.5.3.2.4	liststest	12-42
12.5.3.2.5	report.....	12-42
12.5.3.2.6	run.....	12-43
12.5.3.2.7	status	12-44
12.5.4	Managing the Health Test Framework.....	12-44
12.5.4.1	Creating a Repository and Registering Health Test Framework Tests	12-44
12.5.4.2	Rebuilding the Health Test Framework Indexes	12-45
12.5.5	Running Health Test Framework Diagnostic Tests.....	12-45
12.5.6	Searching for Health Test Framework Diagnostic Tests.....	12-46
12.5.7	Retrieving a Description of a Health Test Framework Test	12-46
12.5.8	Listing Health Test Framework Test Runs	12-46

12.5.9	Generating Health Test Framework Reports.....	12-47
--------	---	-------

Part VI Advanced Administration

13 Managing the Metadata Repository

13.1	Understanding a Metadata Repository.....	13-1
13.2	Creating a Database-Based Metadata Repository	13-2
13.3	Managing the MDS Repository.....	13-2
13.3.1	Understanding the MDS Repository.....	13-3
13.3.1.1	Databases Supported by MDS.....	13-4
13.3.1.2	Understanding MDS Operations	13-5
13.3.2	Registering and Deregistering a Database-Based MDS Repository.....	13-6
13.3.2.1	Registering a Database-Based MDS Repository	13-7
13.3.2.1.1	Registering a Database-Based MDS Repository Using Fusion Middleware# Control	13-7
13.3.2.1.2	Registering a Database-Based MDS Repository Using WLST.....	13-9
13.3.2.2	Adding or Removing Servers Targeted to the MDS Repository	13-9
13.3.2.3	Deregistering a Database-Based MDS Repository.....	13-10
13.3.2.3.1	Deregistering a Database-Based MDS Repository Using Fusion # Middleware Control.....	13-10
13.3.2.3.2	Deregistering a Database-Based MDS Repository Using WLST.....	13-10
13.3.3	Registering and Deregistering a File-Based MDS Repository.....	13-10
13.3.3.1	Creating and Registering a File-Based MDS Repository	13-11
13.3.3.2	Deregistering a File-Based MDS Repository	13-12
13.3.4	Changing the System Data Source	13-12
13.3.5	Using System MBeans to Manage an MDS Repository	13-12
13.3.6	Viewing Information About an MDS Repository	13-13
13.3.6.1	Viewing Information About an MDS Repository Using Fusion Middleware Control	13-13
13.3.6.2	Viewing Information About an MDS Repository Using System MBeans	13-14
13.3.7	Configuring an Application to Use a Different MDS Repository or Partition	13-14
13.3.7.1	Cloning a Partition	13-15
13.3.7.2	Creating a New Partition and Reassociating the Application to It.....	13-17
13.3.8	Moving Metadata from a Source System to a Target System	13-17
13.3.8.1	Transferring Metadata Using Fusion Middleware Control	13-18
13.3.8.2	Transferring Metadata using WLST	13-19
13.3.9	Moving from a File-Based Repository to a Database-Based Repository	13-20
13.3.10	Deleting a Metadata Partition from a Repository.....	13-20
13.3.10.1	Deleting a Metadata Partition Using Fusion Middleware Control.....	13-21
13.3.10.2	Deleting a Metadata Partition Using WLST	13-21
13.3.11	Purging Metadata Version History.....	13-21
13.3.11.1	Purging Metadata Version History Using Fusion Middleware Control.....	13-21
13.3.11.2	Purging Metadata Version History Using WLST	13-22
13.3.11.3	Enabling Auto-Purge	13-22
13.3.12	Managing Metadata Labels in the MDS Repository.....	13-22
13.3.12.1	Creating Metadata Labels.....	13-23
13.3.12.2	Listing Metadata Labels.....	13-23

13.3.12.3	Promoting Metadata Labels.....	13-23
13.3.12.4	Purging Metadata Labels.....	13-24
13.3.12.4.1	Purging Metadata Labels Using Fusion Middleware Control	13-24
13.3.12.4.2	Purging Metadata Labels Using WLST	13-25
13.3.12.5	Deleting Metadata Labels.....	13-26
13.4	Managing Metadata Repository Schemas	13-26
13.4.1	Changing Metadata Repository Schema Passwords	13-26
13.4.2	Changing the Character Set of the Metadata Repository.....	13-26
13.5	Purging Data.....	13-27
13.5.1	Purging Oracle Infrastructure Web Services Data.....	13-28

14 Changing Network Configurations

14.1	Changing the Network Configuration of Oracle Fusion Middleware.....	14-1
14.1.1	Changing the Network Configuration of an Administration Server.....	14-1
14.1.2	Changing the Network Configuration of a Managed Server.....	14-2
14.1.3	Changing the Network Configuration of Oracle HTTP Server	14-3
14.2	Changing the Network Configuration of a Database.....	14-3
14.3	Moving Between On-Network and Off-Network.....	14-6
14.3.1	Moving from Off-Network to On-Network (Static IP Address).....	14-6
14.3.2	Moving from Off-Network to On-Network (DHCP)	14-6
14.3.3	Moving from On-Network to Off-Network (Static IP Address).....	14-6
14.4	Changing Between a Static IP Address and DHCP	14-6
14.4.1	Changing from a Static IP Address to DHCP	14-7
14.4.2	Changing from DHCP to a Static IP Address.....	14-7
14.5	Using IPv6.....	14-7
14.5.1	Configuring Oracle HTTP Server for IPv6.....	14-7

Part VII Advanced Administration: Backup and Recovery

15 Introducing Backup and Recovery

15.1	Understanding Oracle Fusion Middleware Backup and Recovery.....	15-1
15.1.1	Impact of Administration Server Failure	15-2
15.1.2	Managed Server Independence (MSI) Mode	15-2
15.1.3	Configuration Changes in Managed Servers.....	15-2
15.2	Oracle Fusion Middleware Directory Structure.....	15-3
15.3	Overview of the Backup Strategies	15-3
15.3.1	Types of Backups	15-4
15.3.2	Backup Artifacts.....	15-4
15.3.3	Recommended Backup Strategy.....	15-5
15.4	Overview of Recovery Strategies.....	15-6
15.4.1	Types of Recovery.....	15-7
15.4.2	Recommended Recovery Strategies.....	15-7
15.5	Backup and Recovery Recommendations for Oracle Fusion Middleware # Components.....	15-7
15.5.1	Backup and Recovery Recommendations for Oracle WebLogic Server.....	15-7
15.5.1.1	Backup and Recovery Recommendations for Oracle WebLogic Server.....	15-8
15.5.1.2	Backup and Recovery Recommendations for Oracle WebLogic Server JMS ..	15-8

15.5.2	Backup and Recovery Recommendations for Oracle JRF Installations	15-10
15.5.2.1	Backup and Recovery Recommendations for Oracle Web Services Manager	15-10
15.5.2.2	Backup and Recovery Recommendations for Oracle Platform Security # Services.....	15-11
15.5.3	Backup and Recovery Recommendations for Web Tier Installations.....	15-11
15.5.3.1	Backup and Recovery Recommendations for Oracle HTTP Server.....	15-11
15.5.4	Backup and Recovery Recommendations for Oracle Data Integrator	15-12
15.6	Assumptions and Restrictions	15-12

16 Backing Up Your Environment

16.1	Overview of Backing Up Your Environment	16-1
16.2	Limitations and Restrictions for Backing Up Data	16-2
16.3	Performing a Backup	16-3
16.3.1	Performing a Full Offline Backup	16-3
16.3.2	Performing an Online Backup of Run-Time Artifacts.....	16-4
16.3.3	Backing Up Windows Registry Entries	16-5
16.4	Creating a Record of Your Oracle Fusion Middleware Configuration.....	16-5

17 Recovering Your Environment

17.1	Overview of Recovering Your Environment	17-1
17.2	Recovering After Data Loss, Corruption, Media Failure, or Application Malfunction	17-2
17.2.1	Recovering the Oracle Home	17-2
17.2.2	Recovering an Oracle WebLogic Server Domain.....	17-3
17.2.3	Recovering a Standalone Domain	17-3
17.2.4	Recovering the Administration Server Configuration.....	17-4
17.2.5	Recovering a Managed Server	17-4
17.2.5.1	Recovering a Managed Server When It Cannot Be Started.....	17-5
17.2.5.2	Recovering a Managed Server When It Does Not Function Correctly.....	17-6
17.2.6	Recovering Components.....	17-7
17.2.6.1	Recovering a Component That Is Not Functioning Properly	17-7
17.2.6.2	Recovering a Component After Cluster Configuration Change	17-8
17.2.7	Recovering a Cluster	17-8
17.2.7.1	Recovering a Cluster After Deletion or Cluster-Level Configuration Changes	17-8
17.2.7.2	Recovering a Cluster After Membership Is Mistakenly Modified	17-9
17.2.8	Recovering Applications.....	17-10
17.2.8.1	Recovering Application Artifacts.....	17-10
17.2.8.2	Recovering a Redeployed Application That Is No Longer Functional.....	17-11
17.2.8.3	Recovering an Undeployed Application.....	17-11
17.2.8.4	Recovering a Composite Application.....	17-11
17.2.9	Recovering a Database	17-11
17.3	Recovering After Loss of Host	17-12
17.3.1	Recovering After Loss of Oracle WebLogic Server Domain Host.....	17-12
17.3.2	Recovering After Loss of Standalone Domain Host.....	17-12
17.3.2.1	Recovering a Standalone Domain to the Same Host.....	17-12
17.3.2.2	Recovering a Standalone Domain to a Different Host.....	17-13
17.3.3	Recovering After Loss of Administration Server Host	17-14

17.3.3.1	Recovering the Administration Server to the Same Host.....	17-14
17.3.3.2	Recovering the Administration Server to a Different Host.....	17-14
17.3.4	Recovering After Loss of Managed Server Host.....	17-16
17.3.4.1	Recovering a Managed Server to the Same Host.....	17-16
17.3.4.2	Recovering a Managed Server to a Different Host.....	17-17
17.3.5	Recovering After Loss of Component Host.....	17-19
17.3.5.1	Recovering a Java Component to the Same Host.....	17-20
17.3.5.2	Recovering a Java Component to a Different Host.....	17-20
17.3.5.3	Recovering a System Component to the Same Host.....	17-20
17.3.5.4	Recovering a System Component to a Different Host.....	17-20
17.3.5.5	Recovering Web Tier Components to a Different Host.....	17-20
17.3.5.5.1	Recovering Oracle HTTP Server in a Standalone Domain to a Different # Host.....	17-20
17.3.5.5.2	Recovering Oracle HTTP Server in a WebLogic Server Domain to a # Different Host.....	17-21
17.3.5.6	Recovering Oracle Data Integrator to a Different Host.....	17-21
17.3.5.6.1	Recovering Oracle Data Integrator Repository.....	17-21
17.3.5.6.2	Recovering Oracle Data Integrator Agents to a Different Host.....	17-22
17.3.6	Additional Actions for Recovering Entities After Loss of Host.....	17-22
17.3.6.1	Recovering Fusion Middleware Control to a Different Host.....	17-23
17.3.6.2	Modifying the mod_wl_ohs.conf File.....	17-23
17.3.6.3	Creating a New Machine for Certain Components.....	17-23
17.3.6.4	Updating Oracle Inventory.....	17-24
17.3.6.5	Recovering the Windows Registry.....	17-24
17.3.7	Recovering After Loss of Database Host.....	17-25

Part VIII Advanced Administration: Expanding Your Environment

18 Scaling Up Your Environment

18.1	Overview of Scaling Up Your Environment.....	18-1
18.2	Extending a Domain to Support Additional Components.....	18-2
18.3	Adding Additional Managed Servers to a Domain.....	18-3
18.3.1	Applying Oracle JRF Template to a Managed Server or Cluster.....	18-4
18.4	Creating Clusters.....	18-6
18.5	Copying an Oracle Home or Component.....	18-6

19 Moving from a Test to a Production Environment

19.1	Introduction to Moving Oracle Fusion Middleware Components.....	19-1
19.2	Planning for Moving Your Environment.....	19-2
19.2.1	Introduction to the Movement Scripts.....	19-2
19.2.2	Checking the Source Environment.....	19-3
19.2.3	Preparing the Target Environment.....	19-4
19.2.4	Limitations in Moving from Source to Target.....	19-5
19.2.5	Overview of Procedures for Moving from a Source to a Target Environment.....	19-6
19.3	Common Procedures for Moving to a Target Environment.....	19-8
19.3.1	Installing the Database on the Target Environment.....	19-9
19.3.2	Moving the Oracle Home and the Binary Files Using the Scripts.....	19-9

19.3.3	Moving the Oracle Home and Binary Files Using Storage-Level Cloning Tools..	19-11
19.3.4	Moving the Configuration of a WebLogic Server Domain.....	19-12
19.3.5	Moving the Configuration of a Standalone Domain	19-14
19.3.6	Moving the Configuration of Node Manager.....	19-15
19.3.7	Configuring Users and Groups	19-16
19.3.8	Starting Managed Servers and Components	19-17
19.4	Additional Steps for Certain Components.....	19-17
19.4.1	Additional Steps for Moving Oracle Data Integrator.....	19-17
19.5	Incrementally Moving Artifacts.....	19-19
19.6	Moving Distributed Topologies.....	19-19
19.6.1	Considerations with a Multiple Host Environment	19-19
19.6.2	Considerations in Moving to and from an Oracle RAC Environment	19-19
19.7	Recovering from Test to Production Errors	19-20

Part IX Appendixes

A Movement Scripts and Move Plans

A.1	Understanding the Movement Scripts.....	A-1
A.1.1	Movement Scripts Syntax.....	A-3
A.1.1.1	copyBinary Script	A-4
A.1.1.2	pasteBinary Script	A-5
A.1.1.3	copyConfig Script for Oracle WebLogic Server Domains	A-7
A.1.1.4	copyConfig Script for Standalone Domains	A-10
A.1.1.5	copyConfig Script for Node Manager	A-11
A.1.1.6	extractMovePlan Script.....	A-12
A.1.1.7	pasteConfig Script for Oracle WebLogic Server Domains	A-13
A.1.1.8	pasteConfig Script for Standalone Domains	A-15
A.1.1.9	pasteConfig Script for Node Manager.....	A-16
A.1.1.10	obfuscatePassword Script.....	A-17
A.2	Modifying Move Plans.....	A-17
A.2.1	Locating configGroup Elements.....	A-18
A.2.2	Move Plan Properties	A-19

B Oracle Fusion Middleware Command-Line Tools

C URLs for Components

D Port Numbers

D.1	Port Numbers by Component.....	D-1
D.2	Port Numbers (Sorted by Number).....	D-2

E Using Oracle Fusion Middleware Accessibility Options

E.1	Install and Configure Java Access Bridge (Windows Only).....	E-1
E.2	Enabling Fusion Middleware Control Accessibility Mode.....	E-1
E.2.1	Making HTML Pages More Accessible	E-1

E.2.2	Viewing Text Descriptions of Fusion Middleware Control Charts.....	E-2
E.3	Fusion Middleware Control Keyboard Navigation.....	E-3

F Viewing Release Numbers

F.1	Release Number Format	F-1
F.2	Viewing the Software Inventory and Release Numbers	F-2
F.2.1	Viewing Oracle Fusion Middleware Installation Release Numbers	F-2
F.2.2	Viewing Oracle WebLogic Server Release Numbers	F-3
F.2.3	Viewing Component Release Numbers	F-3
F.2.4	Viewing Metadata Repository Release Numbers	F-3
F.2.5	Viewing Schema Release Numbers.....	F-3

G orapki

G.1	Using the orapki Utility for Certificate Validation and CRL Management	G-1
G.1.1	orapki Overview	G-2
G.1.1.1	orapki Syntax	G-2
G.1.1.2	Environment Setup for orapki.....	G-2
G.1.2	Displaying orapki Help	G-2
G.1.3	Creating Signed Certificates for Testing Purposes	G-3
G.1.4	Managing Oracle Wallets with the orapki Utility.....	G-3
G.1.4.1	Creating and Viewing Oracle Wallets with orapki	G-3
G.1.4.2	Adding Certificates and Certificate Requests to Oracle Wallets with orapki	G-4
G.1.4.3	Exporting Certificates and Certificate Requests from Oracle Wallets with # orapki.....	G-5
G.1.5	Managing Certificate Revocation Lists (CRLs) with orapki Utility	G-5
G.1.5.1	About Certificate Validation with Certificate Revocation Lists	G-5
G.1.5.1.1	What CRLs Should You Use?	G-6
G.1.5.1.2	How CRL Checking Works.....	G-6
G.1.5.2	Certificate Revocation List Management	G-6
G.1.5.2.1	Renaming CRLs with a Hash Value for Certificate Validation	G-7
G.1.6	orapki Utility Commands Summary	G-7
G.1.6.1	orapki cert create	G-8
G.1.6.1.1	Purpose	G-8
G.1.6.1.2	Syntax.....	G-8
G.1.6.2	orapki cert display	G-8
G.1.6.2.1	Purpose	G-8
G.1.6.2.2	Syntax.....	G-8
G.1.6.3	orapki crl create	G-8
G.1.6.3.1	Purpose	G-8
G.1.6.3.2	Syntax.....	G-8
G.1.6.4	orapki crl hash.....	G-9
G.1.6.4.1	Purpose	G-9
G.1.6.4.2	Syntax.....	G-9
G.1.6.5	orapki crl revoke.....	G-9
G.1.6.5.1	Purpose	G-9
G.1.6.5.2	Syntax.....	G-9
G.1.6.6	orapki crl status.....	G-9

G.1.6.6.1	Purpose	G-10
G.1.6.6.2	Syntax.....	G-10
G.1.6.7	orapki crl verify.....	G-10
G.1.6.7.1	Purpose	G-10
G.1.6.7.2	Syntax.....	G-10
G.1.6.8	orapki wallet add.....	G-10
G.1.6.8.1	Purpose	G-10
G.1.6.8.2	Syntax.....	G-10
G.1.6.9	orapki wallet change_pwd.....	G-11
G.1.6.9.1	Purpose	G-11
G.1.6.9.2	Syntax.....	G-11
G.1.6.10	orapki wallet create	G-11
G.1.6.10.1	Purpose	G-11
G.1.6.10.2	Syntax.....	G-11
G.1.6.11	orapki wallet display.....	G-11
G.1.6.11.1	Purpose	G-11
G.1.6.11.2	Syntax.....	G-12
G.1.6.12	orapki wallet export	G-12
G.1.6.12.1	Purpose	G-12
G.1.6.12.2	Syntax.....	G-12
G.1.6.13	orapki wallet export_trust_chain	G-12
G.1.6.13.1	Purpose	G-12
G.1.6.13.2	Syntax.....	G-12
G.2	Equivalent Features for orapki.....	G-12
G.3	Features of the SSL Configuration Tool.....	G-13

H Troubleshooting Oracle Fusion Middleware

H.1	Diagnosing Oracle Fusion Middleware Problems	H-1
H.2	Common Problems and Solutions.....	H-1
H.2.1	Running out of Data Source Connections	H-1
H.2.2	Using a Different Version of Spring.....	H-2
H.2.3	ClassNotFoundException Errors When Starting Managed Servers	H-2
H.3	Troubleshooting SSL	H-2
H.3.1	Components May Enable All Supported Ciphers	H-3
H.3.2	SSL Certificate Chain Required on Certain Browsers	H-3
H.4	Need More Help?.....	H-3
H.4.1	Using Remote Diagnostic Agent	H-3

List of Figures

6-1	SSL Handshake.....	6-5
6-2	SSL in Oracle Fusion Middleware	6-5
12-1	ADR Directory Structure for Oracle Fusion Middleware	12-5
12-2	Incident Creation Generated by Incident Log Detector	12-7
12-3	Incident Creation Generated by WLDF Watch Notification	12-9
12-4	Flow for Investigating a Problem	12-21
16-1	Decision Flow Chart for Type of Backup	16-2
19-1	Flowchart for Moving Your Environment	19-6
F-1	Example of an Oracle Fusion Middleware Release Number.....	F-1

List of Tables

2-1	Comparing Fusion Middleware Control and WebLogic Server Administration Console	2-2
2-2	Navigating Within Fusion Middleware Control.....	2-7
2-3	WLST Commands for System Components	2-13
6-1	WLST Commands for SSL Configuration	6-23
6-2	WLST Commands for Oracle Wallet Management	6-24
6-3	Parameters in Properties File	6-35
6-4	Default Values of Parameters.....	6-36
7-1	Keystore Types in Oracle Fusion Middleware	7-1
9-1	Tools to Deploy Applications.....	9-2
9-2	MDS Configuration Attributes for Deployed Applications	9-24
11-1	ODL Format Message Fields	11-3
11-2	Log File Location.....	11-4
11-3	Diagnostic Message Types and Level	11-17
11-4	Mapping of Log Levels Among ODL, Oracle WebLogic Server, and Java	11-18
12-1	DiagnosticConfig MBean Attributes for Diagnostic Framework	12-10
12-2	Conditions for the LogDetectionConditions Element	12-14
12-3	Optional arguments for the defaultActions Element	12-14
12-4	Attributes for the ruleCondition Element	12-15
12-5	DiagnosticConfig MBean Operations and Attributes for Problem Suppression # Filters	12-17
12-6	Uncaught Exception Problem Keys.....	12-25
12-7	Diagnostic Dump Actions.....	12-26
12-8	Default Diagnostic Dump Samplings Configuration	12-28
12-9	Health Test Framework dfwhealthtestadminctl.sh Commands.....	12-40
12-10	Health Test Framework dfwhealthtestctl.sh Commands	12-41
13-1	MDS Operations and Required Roles	13-5
13-2	Purging Data Documentation	13-27
17-1	Recovery Procedures for Loss of Host for Particular Components.....	17-20
19-1	Support for Movement Scripts.....	19-3
19-2	Components Requiring Additional Steps for Movement to a Different # Environment	19-17
A-1	Movement Scripts	A-1
A-2	Options for the copyBinary Script	A-5
A-3	Options for the pasteBinary Script	A-6
A-4	Options for the copyConfig Script for a Oracle WebLogic Server Domains.....	A-8
A-5	Options for the copyConfig Script for Standalone Domains.....	A-10
A-6	Options for the copyConfig Script for Node Manager.....	A-12
A-7	Options for the extractMovePlan Script	A-13
A-8	Options for the pasteConfig Script for Oracle WebLogic Server Domains.....	A-14
A-9	Options for the pasteConfig Script for Standalone Domains	A-16
A-10	Options for the pasteConfig Script for Node Manager	A-17
A-11	Move Plan Properties for Components	A-19
A-12	Move Plan Properties for Node Manager	A-20
A-13	Common Move Plan Properties for Java Components	A-21
A-14	Move Plan Properties for Oracle ADF Connections	A-28
A-15	Move Plan Properties for Oracle Coherence.....	A-29
A-16	Move Plan Properties for Oracle User Messaging Service.....	A-30
A-17	Move Plan Properties for Oracle HTTP Server.....	A-32
A-18	Move Plan Properties for Oracle Web Services Manager	A-34
A-19	Move Plan Properties for Oracle Data Integrator	A-36
B-1	Oracle Fusion Middleware Command-Line Tools	B-1
C-1	URLs for Components.....	C-1

D-1	Port Numbers Sorted by Component	D-1
D-2	Port Numbers Sorted by Number	D-2
E-1	Keyboard Navigation for Common Tasks	E-3
E-2	Keyboard Navigation for Topology Viewer	E-4
G-1	Mapping for orapki Features for Wallets and CRLs.....	G-13
G-2	Mapping for orapki Features for Certificates	G-13
G-3	Equivalent Features for the SSL Configuration Tool.....	G-14

Preface

This guide describes how to manage Oracle Fusion Middleware, including how to start and stop Oracle Fusion Middleware, how to change ports, deploy applications, how to back up and recover Oracle Fusion Middleware and how to move your environment from a source environment, such as a test environment to a target environment, such as a production environment.

Audience

This guide is intended for administrators of Oracle Fusion Middleware.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware 12c (12.1.2) documentation set:

- *Understanding Oracle Fusion Middleware*
- *Securing Applications with Oracle Platform Security Services*
- *High Availability Guide*
- *Understanding Oracle WebLogic Server*
- *Tuning Performance*
- *Administering Oracle HTTP Server*
- *Administering Web Services*

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.

What's New in This Guide?

This preface introduces the new and changed administrative features of Oracle Fusion Middleware that are described in this guide, and provides pointers to additional information.

Updates for Release 12c Documentation Refresh, December 2014

This edition of the guide contains bug fixes and other corrections.

New and Changed Features for Release 12c

The following topics introduce the new and changed features of Oracle Fusion Middleware and other significant changes that are described or referred to in this guide, and provides pointers to additional information. This book is the new edition of the formerly titled *Oracle Fusion Middleware Administrator's Guide*.

- Redefining of the Oracle home and elimination of the Middleware home. See "New And Deprecated Terminology for 12c" in *Understanding Oracle Fusion Middleware*.
- OPMN is no longer used in Oracle Fusion Middleware. Instead, system components are managed by the WebLogic Management Framework, which includes WLST, Node Manager and pack and unpack. See "What Is the WebLogic Management Framework" in *Understanding Oracle Fusion Middleware*.
- Support for a "per domain" Node Manager. See "What Is Node Manager?" in *Understanding Oracle Fusion Middleware*.
- Oracle Web Cache is no longer part of Oracle Fusion Middleware.
- Changes in moving from a source to a target environment:
 - Because of the redefining of Oracle home and elimination of Middleware home, some of the parameters to the scripts have changed. See [Section A.1.1](#).
 - Support for moving a standalone domain. See [Section 19.3.5](#).
 - Support for moving the Oracle home and binary files using storage-level cloning tools. See [Section 19.3.3](#).
 - A move plan for Oracle Coherence. See [Table A-15](#).
 - A move plan for Oracle Web Services Manager. See [Table A-18](#).
 - Support for moving Oracle Data Integrator.
- The OPSS Keystore Service is introduced. See [Section 7.1.1.3](#).
- SSL procedures for Oracle WebLogic Server have been updated. See [Section 6.5.1](#).

- Fusion Middleware Control supports cross-component wiring. See [Section 3.3](#).
- Oracle Fusion Middleware has introduced service tables, which provide a way for service providers to publish endpoint information about their services, and clients of these services to query and bind to these services. See [Section 3.1](#).
- Updated procedures for backup and recovery, including procedures for recovering a standalone domain. See [Chapter 17](#).
- Enhanced support for querying diagnostic incidents. See [Section 12.4.2.3](#).
- Support for creating aggregated incidents. See [Section 12.4.6.2](#).
- Support for extended log format for access logs. See [Section 11.1.1](#).

Part I

Understanding Oracle Fusion Middleware

This part provides an overview to Oracle Fusion Middleware and its concepts as they relate to administering Oracle Fusion Middleware.

Part I contains the following chapter:

- [Chapter 1, "Introduction to Oracle Fusion Middleware"](#)

Introduction to Oracle Fusion Middleware

Oracle Fusion Middleware is a comprehensive family of products ranging from application development tools and integration solutions to identity management, collaboration, and business intelligence reporting. This chapter provides an introduction to Oracle Fusion Middleware.

It includes the following sections:

- [What Is Oracle Fusion Middleware?](#)
- [Oracle Fusion Middleware Components](#)

1.1 What Is Oracle Fusion Middleware?

Oracle Fusion Middleware is a collection of standards-based software products that spans a range of tools and services: from Java EE and developer tools, to integration services, identity management, business intelligence, and collaboration. Oracle Fusion Middleware offers complete support for development, deployment, and management.

For information about Oracle Fusion Middleware concepts, see *Understanding Oracle Fusion Middleware*.

1.2 Oracle Fusion Middleware Components

Oracle Fusion Middleware provides the following components:

- Oracle WebLogic Server, an enterprise-ready Java application server that supports the deployment of mission-critical applications in a robust, secure, highly available, and scalable environment.

See Also: *Understanding Oracle WebLogic Server*

- Oracle HTTP Server, which provides a Web listener for Java EE applications and the framework for hosting static and dynamic pages and applications over the Web. Based on the proven technology of the Apache HTTP Server, Oracle HTTP Server includes significant enhancements that facilitate load balancing, administration, and configuration.

See Also: *Administering Oracle HTTP Server*

- Oracle Web Services Manager, which provides a way to centrally define and manage policies that govern Web services operations, including access control (authentication and authorization), reliable messaging, Message Transmission Optimization Mechanism (MTOM), WS-Addressing, and Web services

management. Policies can be attached to multiple Web services, requiring no modification to the existing Web services.

See Also: *Administering Web Services*

- Oracle Platform Security, which provides enterprise product development teams, systems integrators, and independent software vendors (ISVs) with a standards-based, portable, integrated, enterprise-grade security framework for Java Standard Edition (Java SE) and Java Enterprise Edition (Java EE) applications.

Oracle Platform Security provides an abstraction layer in the form of standards-based application programming interfaces (APIs) that insulate developers from security and identity management implementation details. With Oracle Platform Security, developers do not need to know the details of cryptographic key management or interfaces with user repositories and other identity management infrastructures. When you use Oracle Platform Security, in-house developed applications, third-party applications, and integrated applications benefit from the same uniform security, identity management, and audit services across the enterprise.

See Also: *Securing Applications with Oracle Platform Security Services*

Part II

Basic Administration

This part describes basic administration tasks for Oracle Fusion Middleware.

Part II contains the following chapters:

- [Chapter 2, "Getting Started Managing Oracle Fusion Middleware"](#)
- [Chapter 3, "Cross-Component Wiring"](#)
- [Chapter 4, "Starting and Stopping Oracle Fusion Middleware"](#)
- [Chapter 5, "Managing Ports"](#)

Getting Started Managing Oracle Fusion Middleware

When you install Oracle Fusion Middleware, you install the binary files, such as executable files, jar files, and libraries. Then, you use configuration tools to configure the software. This chapter provides information you need to get started managing Oracle Fusion Middleware, including information about the tools you use.

This chapter includes the following sections:

- [Overview of Oracle Fusion Middleware Administration Tools](#)
- [Getting Started Using Oracle Enterprise Manager Fusion Middleware Control](#)
- [Getting Started Using Oracle WebLogic Server Administration Console](#)
- [Getting Started Using the Oracle WebLogic Scripting Tool \(WLST\)](#)
- [Getting Started Using the Fusion Middleware Control MBean Browsers](#)
- [Managing Components](#)
- [Changing the Administrative User Password](#)
- [Configuring Node Manager](#)
- [Basic Tasks for Configuring and Managing Oracle Fusion Middleware](#)

2.1 Overview of Oracle Fusion Middleware Administration Tools

After you install and configure Oracle Fusion Middleware, you can use the graphical user interfaces or command-line tools to manage your environment.

Oracle offers the following primary tools for managing your Oracle Fusion Middleware installations:

- Oracle Enterprise Manager Fusion Middleware Control. See [Section 2.2](#).
- Oracle WebLogic Server Administration Console. See [Section 2.3](#)
- The Oracle Fusion Middleware command-line tools. See [Section 2.4](#).
- The Fusion Middleware Control MBean Browser. See [Section 2.5](#).

Note that you should use these tools, rather than directly editing configuration files, to perform all administrative tasks unless a specific procedure requires you to edit a file. Editing a file may cause the settings to be inconsistent and generate problems.

Both Fusion Middleware Control and Oracle WebLogic Server Administration Console are graphical user interfaces that you can use to monitor and administer your Oracle Fusion Middleware environment. You can install Fusion Middleware Control and the

Administration Console when you install most Oracle Fusion Middleware components.

Note the following:

- If you install a standalone Oracle WebLogic Server, Fusion Middleware Control is not installed. Only the Administration Console is installed.
- If you install Oracle JDeveloper, neither Fusion Middleware Control or the Administration Console are installed. They can be installed if you install Oracle Fusion Middleware Application Developer.

You can perform some tasks with either tool, but for other tasks, you can only use one of the tools. [Table 2–1](#) lists some common tasks and the recommended tool.

Table 2–1 Comparing Fusion Middleware Control and WebLogic Server Administration Console

Task	Tool to Use
Manage Oracle WebLogic Server	Use:
Create additional Managed Servers	Fusion Middleware Control
Clone Managed Servers	WebLogic Server Administration Console
Cluster Managed Servers	Fusion Middleware Control
Start and stop Oracle WebLogic Server	Fusion Middleware Control or WebLogic Server Administration Console
Add users and groups	WebLogic Server Administration Console if using the default embedded LDAP; if using another LDAP server, use the LDAP server's tool
Manage Data Sources	Use:
Create data sources	Fusion Middleware Control or WebLogic Server Administration Console
Create connection pools	Fusion Middleware Control or WebLogic Server Administration Console
Manage JMS Resources	Use:
Create JMS queues	WebLogic Server Administration Console
Configure advanced queuing	WebLogic Server Administration Console
ADF Applications	Use:
Deploy ADF applications	Fusion Middleware Control
Java EE applications	Use:
Deploy Java EE applications	WebLogic Server Administration Console or Fusion Middleware Control
Security	Use:
Configure and manage auditing	Fusion Middleware Control
Configure SSL	WebLogic Server Administration Console for Oracle WebLogic Server Fusion Middleware Control for Java components and system components. See Chapter 6 .
Change passwords	WebLogic Server Administration Console
Manage Components	Use:

Table 2–1 (Cont.) Comparing Fusion Middleware Control and WebLogic Server Administration Console

Task	Tool to Use
View and manage log files	Fusion Middleware Control for most log files WebLogic Server Administration Console for the following logs: <i>DOMAIN_HOME/servers/server_name/logs/access.log</i> <i>DOMAIN_HOME/servers/server_name/data/ldap/log/EmbeddedLDAP.log</i> <i>DOMAIN_HOME/servers/server_name/data/ldap/log/EmbeddedLDAPAccess.log</i>
Change ports	WebLogic Server Administration Console for Oracle WebLogic Server and Java components For some system components, Fusion Middleware Control. See the administration guide for the component.
Manage Oracle HTTP Server	Fusion Middleware Control
Start and stop components	Fusion Middleware Control
Start and stop applications	Fusion Middleware Control

2.2 Getting Started Using Oracle Enterprise Manager Fusion Middleware Control

Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer your domain. It can manage an Oracle WebLogic Server domain with its Administration Server, one or more Managed Servers, clusters, the Oracle Fusion Middleware components that are installed, configured, and running in the domain, and the applications you deploy.

Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the domain, servers, components, and applications. The Fusion Middleware Control home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions—all from your Web browser.

The following topics are discussed in this section:

- [Displaying Fusion Middleware Control](#)
- [Using Fusion Middleware Control Help](#)
- [Navigating Within Fusion Middleware Control](#)
- [Understanding Users and Roles for Fusion Middleware Control](#)
- [Viewing and Managing the WebLogic Domain](#)
- [Viewing and Managing Components](#)

2.2.1 Displaying Fusion Middleware Control

To display Fusion Middleware Control, you enter the Fusion Middleware Control URL, which includes the name of the host and the administration port number assigned during the installation. The following shows the format of the URL:

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

The port number is the port number of the Administration Server. By default, the port number is 7001. The port number is listed in the following file:

```
DOMAIN_HOME/config/config.xml
```

For some installation types, such as Web Tier, if you saved the installation information by clicking Save on the last installation screen, the URL for Fusion Middleware Control is included in the file that is written to disk (by default to your home directory). For other installation types, the information is displayed on the Create Domain screen of the Configuration Wizard when the configuration completes.

To display Fusion Middleware Control:

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

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

The following shows the login page:



2. Enter the Oracle Fusion Middleware administrator user name and password and click **Login**.

The default user name for the administrator user is `weblogic`. This is the account you can use to log in to Fusion Middleware Control for the first time. The

password is the one you supplied during the installation of Oracle Fusion Middleware.

2.2.2 Using Fusion Middleware Control Help

At any time while using the Fusion Middleware Control Console, you can click **Help** at the top of the page to get more information. From that menu, You can select the following: in the Help window to browse the list of help topics, or click **Search** to search for a particular word or phrase.

- **Help Contents**, which lists the contents of Help.
- **Page Reference**, which provides context-sensitive help for the current page.
- **How Do I?**, which links to tutorial information in the documentation.
- **Documentation Library**, which links to the library on the Oracle Technology Network.
- **User Forums**, which links to Discussion Forums on the Oracle Technology Network.
- **Oracle Technology Network**, which links to the Oracle Technology Network.

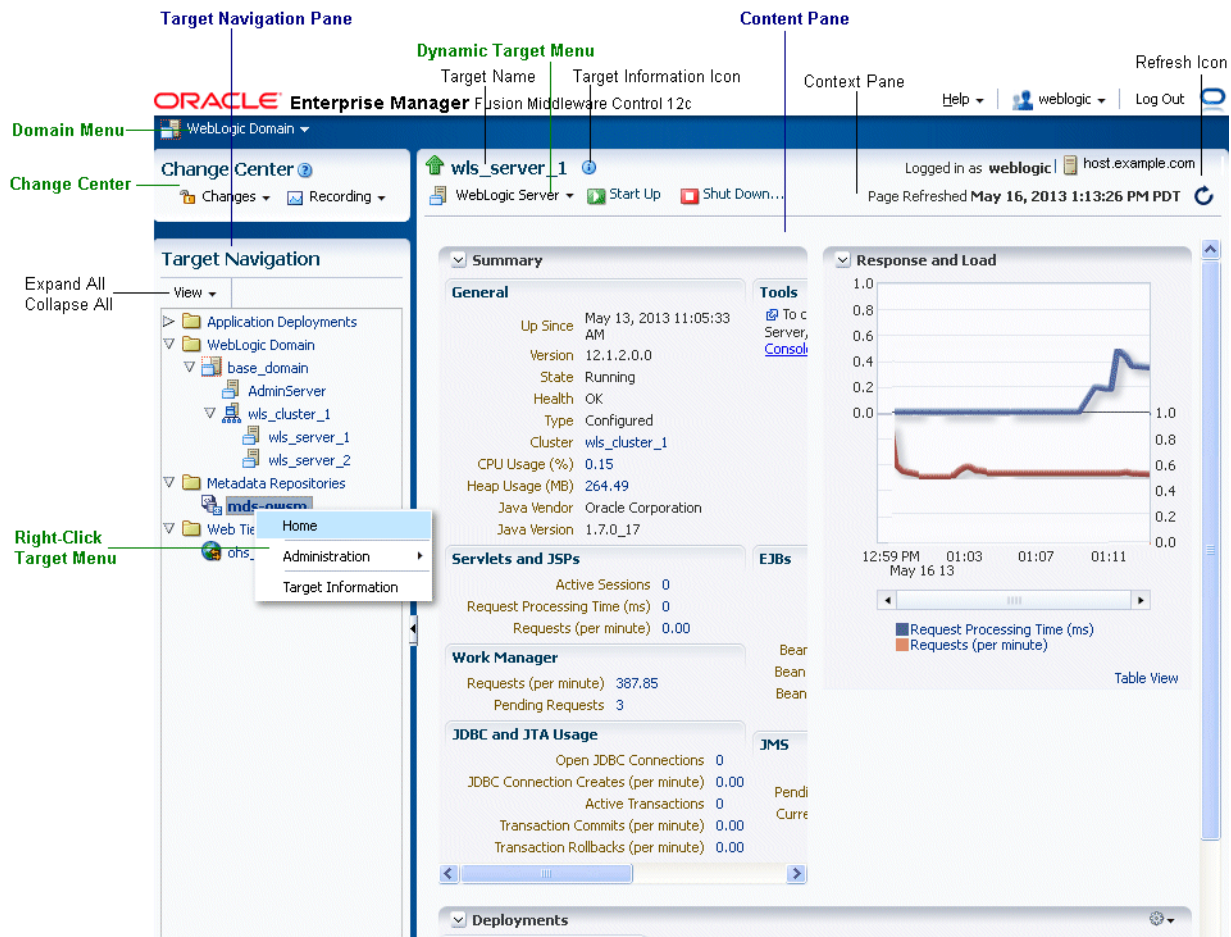
2.2.3 Navigating Within Fusion Middleware Control

Fusion Middleware Control displays the target navigation pane on the left and the content pane on the right. For example, when you first log in to Fusion Middleware Control, the domain home page is displayed on the right.

From the target navigation pane, you can expand the tree and select an Oracle WebLogic Server domain, an Oracle WebLogic Server Managed Server, a component, an application, or a Metadata Repository.

When you select a target, such as a Managed Server or a component, the target's home page is displayed in the content pane and that target's menu is displayed at the top of the page, in the context pane. For example, if you select a Managed Server, the WebLogic Server menu is displayed. You can also view the menu for a target by right-clicking the target in the navigation pane.

The following figure shows the target navigation pane and the home page of an Managed Server. Because a Managed Server was selected, the dynamic target menu listed in the context pane is the WebLogic Server menu.



In the preceding figure, the following items are called out:

- **WebLogic Domain Menu** provides a list of operations that you can perform on the domain. The WebLogic Domain menu is always available.
- **Change Center** shows the changes made and allows you to lock and edit the configuration, release the configuration, activate changes, undo all changes, and change preferences. It also allows you to start and stop recording your session, and view the recording.
- **Target Navigation Pane** lists all of the targets in the domain in a navigation tree.
- **Content Pane** shows the current page for the target. When you first select a target, that target's home page is displayed.
- **Dynamic Target Menu** provides a list of operations that you can perform on the currently selected target. The menu that is displayed depends on the target you select. The menu for a specific target contains the same operations as those in the **Right-Click Target Menu**.
- **Right-Click Target Menu** provides a list of operations that you can perform on the currently selected target. The menu is displayed when you right-click the target name in the target navigation pane. In the figure, even though the WebLogic Server is selected and its home page is displayed, the right-click target menu displays the operations for a metadata repository because the user has right-clicked the metadata repository.

The menu for a specific target contains the same operations as those in the **Dynamic Target Menu**.

- **Target Name** is the name of the currently selected target.
- **Target Information Icon** provides information about the target. For example, for a domain, it displays the target name, the version, and the domain home.
- **Context Pane** provides the name of the target, the name of the current user, the host name, and the time of the last page refresh, as well as the Refresh icon.
- **View** lets you expand or collapse the navigation tree.
- **Help** provides help for Fusion Middleware Control.
- The *username* lets you change accessibility preferences.
- Logout logs you out of Fusion Middleware Control.
- **Refresh** indicates when the page is being refreshed. Click it to refresh a page with new data. (Refreshing the browser window refreshes the page but does not retrieve new data.)
- **Log out** logs you out for Fusion Middleware Control.

In addition, from Fusion Middleware Control, from the home pages of targets such as the Administration Server or Managed Servers, you can access the WebLogic Server Administration Console.

[Table 2–2](#) describes some common ways you can navigate within Fusion Middleware Control.

Table 2–2 Navigating Within Fusion Middleware Control

To:	Take This Action:
View all of the targets in the domain	From the View menu, select Expand All .
Operate on the domain	Select the WebLogic Domain menu , which is always available at the top left of Fusion Middleware Control.
Operate on a target	Right-click the target in the target navigation pane . The target menu is displayed. Alternatively, you can select the target and use the dynamic target menu in the context pane.
Return to the target's home page	Click the target name at the top left-hand corner of the context pane .
Refresh a page with new data	Click the Refresh icon in the top right of the context pane .
Return to a previous page	Click the breadcrumbs, which appear below the context pane. The breadcrumbs appear when you drill down in a target. For example, choose Logs from the WebLogic Server menu, then View Log Messages. Select a log file and click View Log File. The breadcrumbs show: Log Messages > Log Files > View Log File: <i>logfile_name</i>
View the host on which the target is running	Select the target in the target navigation pane and view the host name in the target's context pane . You can also view the host name by clicking the Target Information icon.
View a server log file	Right-click the server name in the target navigation pane . Choose Logs , and then View Log Messages to see a summary of log messages and to search log files.

2.2.4 Understanding Users and Roles for Fusion Middleware Control

To access Fusion Middleware Control and perform tasks, you must have the appropriate role. Fusion Middleware Control uses the Oracle WebLogic Server security realm and the roles defined in that realm. If a user is not granted one of these roles, the user cannot access Fusion Middleware Control.

Each role defines the type of access a user has. For example, a user with the role Admin has full privileges. A user with the role Operator has privileges to perform essential day-to-day operations. A user with the role Monitor has privileges only to view the configuration.

For more information, see "Users, Groups, and Security Roles" in the *Securing Resources Using Roles and Policies for Oracle WebLogic Server*.

2.2.5 Viewing and Managing the WebLogic Domain

When you log in to Fusion Middleware Control, the first page you see is the domain home page. You can also view this page at any time by selecting Home in the WebLogic Domain menu.

The following figure shows the domain home page:

The screenshot displays the Oracle Enterprise Manager Fusion Middleware Control interface for a WebLogic Domain named 'base_domain'. The page is titled 'base_domain' and shows the user is logged in as 'weblogic'. The page was refreshed on May 17, 2013, at 12:57:36 PM PDT. The interface is divided into several sections:

- Change Center:** Shows 'Changes' and 'Recording' options.
- Target Navigation:** A tree view on the left showing the hierarchy: Application Deployments > WebLogic Domain > base_domain > AdminServer > wls_cluster_1 > Metadata Repositories > Web Tier.
- Summary:**
 - General:** Administration Server: AdminServer; Administration Server Host: host.example.com; Administration Server Listen Port: 7001.
 - Tools:** A checkbox for 'To configure and manage this WebLogic Domain, use the WebLogic Server Administration Console' is checked.
 - Servers:** A large green circle indicates 100% of servers are up.
 - Table:**

Name	Status	Type	Host	Clust
AdminServer	Up	Configur...
wls_server_1	Up	Configur...	...	wls_...
wls_server_2	Up	Configur...	...	wls_...
 - Clusters:** A table showing one cluster:

Name	Servers	Cluster Address	Cluster Messaging Mode
wls_cluster_1	2		Unicast
 - Deployments:** A pie chart shows 83% of deployments are up (green) and 17% are down (red). Below the chart is a table with columns: Name, Status, Deployed C.
 - Web Services:** A table with columns: Name, Server, Applica. It shows 'No Web Services Found'.
 - Oracle WebLogic Domain Resource Center:** A section with a 'Before You Begin' heading and links for 'What is a WebLogic Domain?' and 'Manage Oracle WebLogic Server with Fusion'.

The WebLogic Domain menu is displayed at the top of the page. From this menu, you can monitor and configure the domain.

The WebLogic Domain menu is always displayed, even if you have selected other entities.

You can view the routing topology by selecting **Routing Topology** from the WebLogic Domain menu. The Topology Viewer provides you with a high-level view of the topology, including Managed Servers, deployed applications, and the routing configuration. See [Section 10.3](#) for information about using the Topology Viewer.

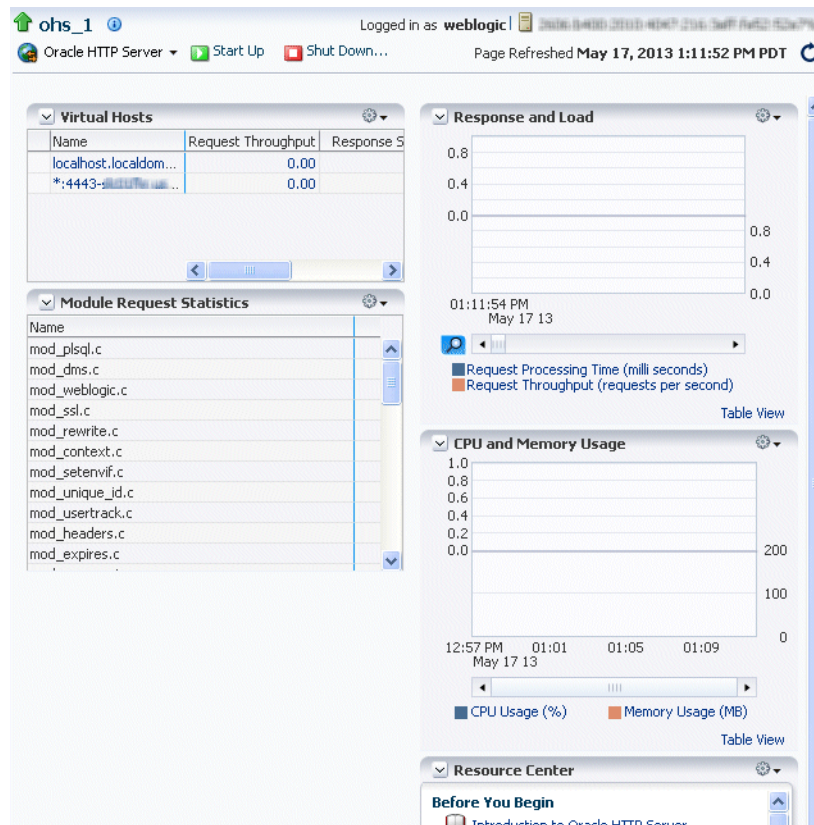
2.2.6 Viewing and Managing Components

From the target navigation pane, you can drill down to view and manage the components in your domain.

For example, to view and manage Oracle HTTP Server, take the following steps:

1. In the target navigation pane, expand **Web Tier**.
2. Select the Oracle HTTP Server instance, for example, ohs1.

The home page for the Oracle HTTP Server ohs1 is displayed, as shown in the following figure:



3. From the HTTP Server menu, you can perform many administrative tasks, such as starting, stopping, and monitoring Oracle HTTP Server.

For more information about monitoring components, see [Section 10.1.4](#).

2.3 Getting Started Using Oracle WebLogic Server Administration Console

Oracle WebLogic Server Administration Console is a Web browser-based, graphical user interface that you use to manage an Oracle WebLogic Server domain. It is accessible from any supported Web browser with network access to the Administration Server.

Use the Administration Console to:

- Configure, start, and stop WebLogic Server instances
- Configure WebLogic Server clusters
- Configure WebLogic Server services, such as database connectivity (JDBC) and messaging (JMS)
- Configure security parameters, including creating and managing users, groups, and roles
- Configure and deploy Java EE applications
- Monitor server and application performance
- View server and domain log files
- View application deployment descriptors
- Edit selected run-time application deployment descriptor elements

2.3.1 Displaying the Oracle WebLogic Server Administration Console

To display the Administration Console:

1. Enter the following URL in a browser:

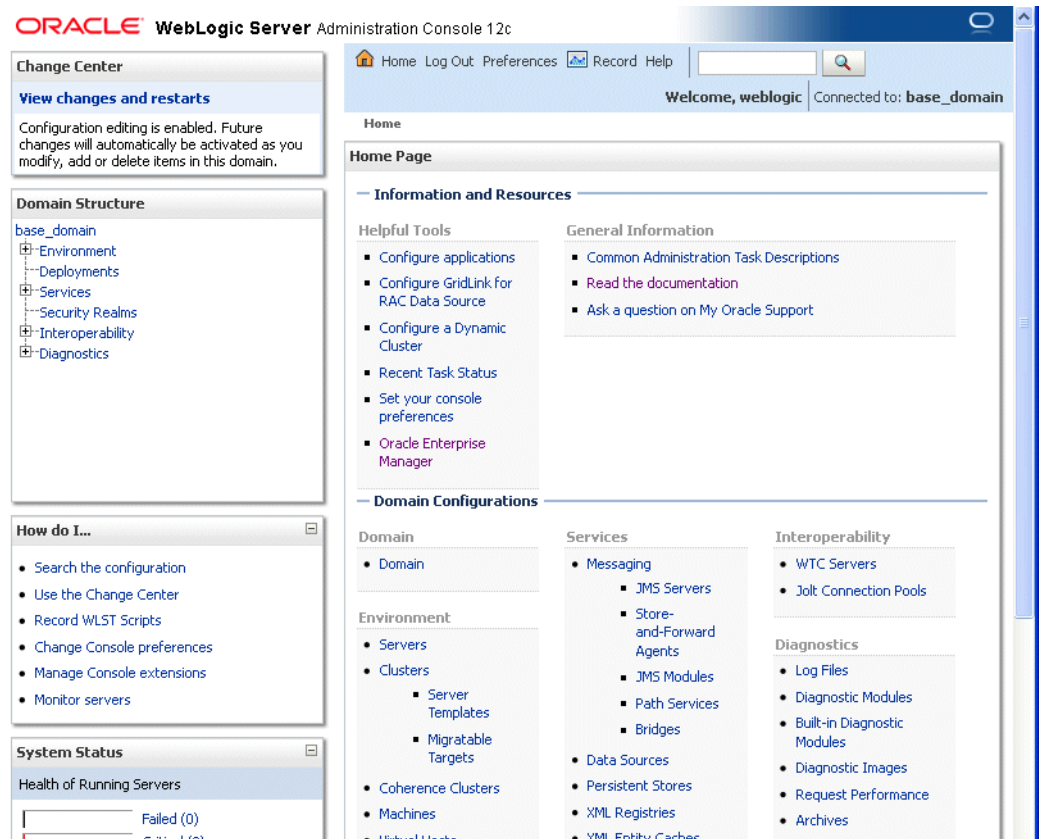
```
http://hostname:port_number/console
```

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

The login page is displayed.

2. Log in using the user name and password supplied during installation or another administrative user that you created.

Oracle WebLogic Server Administration Console is displayed as shown in the following figure:



Alternatively, you can access the Administration Console from Fusion Middleware Control, from the home pages of targets such as the Administration Server or Managed Servers.

2.3.2 Locking the WebLogic Server Configuration

Before you make configuration changes, lock the domain configuration, so you can make changes to the configuration while preventing other accounts from making changes during your edit session. To lock the domain configuration:

1. Locate the Change Center in the upper left of the Administration Console screen.
2. Click **Lock & Edit** to lock the configuration edit hierarchy for the domain.

As you make configuration changes using the Administration Console, you click **Save** (or in some cases **Finish**) on the appropriate pages. This does not cause the changes to take effect immediately. The changes take effect when you click **Activate Changes** in the Change Center. At that point, the configuration changes are distributed to each of the servers in the domain. If the changes are acceptable to each of the servers, then they take effect. If any server cannot accept a change, then all of the changes are rolled back from all of the servers in the domain. The changes are left in a pending state; you can then either edit the pending changes to resolve the problem or revert to the previous configuration.

You can also lock the configuration by using the WLST command, `startEdit`:

```
startEdit()
```

For more information about the `startEdit` command and the `stopEdit` command, which releases locks, see "startEdit" and "stopEdit" in the *WLST Command Reference for WebLogic Server*.

2.4 Getting Started Using the Oracle WebLogic Scripting Tool (WLST)

The Oracle WebLogic Scripting Tool (WLST) is a command-line scripting environment that you can use to create, manage, and monitor Oracle WebLogic Server domains. It is based on the Java scripting interpreter, Jython. In addition to supporting standard Jython features such as local variables, conditional variables, and flow-control statements, WLST provides a set of scripting functions (commands) that are specific to WebLogic Server. You can extend the WebLogic scripting language to suit your needs by following the Jython language syntax.

The following topics describe using WLST to manage Oracle Fusion Middleware components:

- [Using WLST with Java Components](#)
- [Using Custom WLST Commands](#)
- [Using WLST Commands with System Components](#)

2.4.1 Using WLST with Java Components

You can use WLST commands with Java components. You can use them in the following ways:

- Interactively, on the command line
- In script mode, supplied in a file
- Embedded in Java code

For example, to invoke WLST interactively, and connect to the WebLogic Server, use the following commands:

```
WL_HOME/server/bin/setWLSEnv.sh
WL_HOME/common/bin/wlst.sh
connect('username',#'password',#'localhost:7001')
```

To display information about WLST commands and variables, enter the help command. For example, to display a list of categories for online commands, enter the following:

```
wls:/base_domain/serverConfig>##help('online')
###help('activate')#####Activate#the#changes.
###help('addListener')#####Add#a#JMX#listener#to#the#specified#MBean.
###help('adminHome')#####Administration#MBeanHome.
###help('cancelEdit')#####Cancel#an#edit#session.
###help('cd')#####Navigate#the#hierarchy#of#beans.
###help('cmo')#####Current#Management#Object.
#
#
#
```

To monitor the status, you use the WLST `state` command, using the following format:

```
state(name,#type)
```

For example to get the status of the Managed Server `wls_server1`, use the following command:

```
wls:/WLS_domain/serverConfig>#state('wls_server1',#'Server')
Current#state#of#'wls_server1'#: #RUNNING
```

For more information about WLST, see *WLST Command Reference for WebLogic Server*.

2.4.2 Using Custom WLST Commands

Many components and services, such as Oracle Application Development Framework, Diagnostic Framework, MDS, and SSL and logging, provide custom WLST commands.

To use those custom commands, you must invoke the WLST script from the Oracle Common home. Do not use the WLST script in the WebLogic Server home.

- Oracle Application Development Framework
- Oracle Metadata Services
- Diagnostic Framework
- Dynamic Monitoring Service (DMS)
- Logging
- Secure Sockets Layer (SSL)
- Oracle JRF
- Oracle Web Services
- Oracle Web Services Manager
- System components, such as Oracle HTTP Server

2.4.3 Using WLST Commands with System Components

You can use WLST commands with system components. The following component is a system component:

- Oracle HTTP Server

For system components, you can only use the WLST commands listed in [Table 2–3](#). See *WLST Command Reference for WebLogic Server* for information about whether a command can be invoked in online or offline mode.

Table 2–3 WLST Commands for System Components

WLST Command	Description	For more information:
create	Creates an instance of the system component with defaults	The create command in <i>WLST Command Reference for WebLogic Server</i>
displayLogs	Displays the messages in a log file	The displayLogs command in <i>WLST Command Reference for Infrastructure Components</i> and Section 11.3.1.2
displayMetricTableNames	Displays the names of the DMS metric tables	The displayMetricTableNames command in <i>WLST Command Reference for Infrastructure Components</i>
displayMetricTables	Displays the contents of the DMS metric tables	The displayMetricTables command in <i>WLST Command Reference for Infrastructure Components</i>
dumpMetrics	Displays the available DMS metrics.	The dumpMetrics command in <i>WLST Command Reference for Infrastructure Components</i>

Table 2–3 (Cont.) WLST Commands for System Components

WLST Command	Description	For more information:
listLogs	Lists the log files.	The listLogs command in <i>WLST Command Reference for Infrastructure Components</i> and Section 11.3.1.2
nmKill	Shuts down and instance	The nmkill command in <i>WLST Command Reference for WebLogic Server</i> and Section 4.3.2.2 .
nmServerStatus	Returns the status on an instance	The nmServerStatus command in <i>WLST Command Reference for WebLogic Server</i>
nmStart	Starts an instance	The nmStart command in <i>WLST Command Reference for WebLogic Server</i> and Section 4.3.2.2 .
shutdown	Stops a system component instance.	The shutdown command in <i>WLST Command Reference for WebLogic Server</i> and Section 4.3.2.2 .
start	Starts a system component instance.	The start command in <i>WLST Command Reference for WebLogic Server</i> and Section 4.3.2.2 .
state	Returns the state of a system component instance	The state command in <i>WLST Command Reference for WebLogic Server</i> and Section 2.4

To use these commands, you must invoke the WLST script from the Oracle common home in which the component has been installed. Do not use the WLST script in the WebLogic Server home. The script is located at:

```
(UNIX) #ORACLE_HOME/oracle_common/common/bin/wlst.sh
(Windows) #ORACLE_HOME\oracle_common\common\bin\wlst.cmd
```

To monitor the status of a system component, you use the WLST `state` command, using the following format:

```
state(component_name, #SystemComponent)
```

In online mode, you can use the `cmo` variable to invoke MBean operations that provide even more functionality. For more information about the `cmo` variable, see "Changing the Current Management Object" in *Understanding the WebLogic Scripting Tool*.

2.5 Getting Started Using the Fusion Middleware Control MBean Browsers

A **managed bean** (MBean) is a Java object that represents a JMX manageable resource in a distributed environment, such as an application, a service, a component or a device.

MBeans are defined in the Java EE Management Specification (JSR-77), which is part of Java Management Extensions, or JMX, a set of specifications that allow standard interfaces to be created for managing applications in a Java EE environment. For information about JSR-77, see:

<http://java.sun.com/j2ee/tools/management/>

You can create MBeans for deployment with an application into Oracle WebLogic Server, enabling the application or its components to be managed and monitored through Fusion Middleware Control.

Fusion Middleware Control provides a set of MBean browsers that allow you to browse the MBeans for an Oracle WebLogic Server or for a selected application. You can also perform specific monitoring and configuration tasks from the MBean browser.

The MBeans are organized into three groups: Configuration MBeans, Runtime MBeans, and Application-Defined MBeans.

For more information about MBeans, see "Understanding WebLogic Server MBeans" in *Developing Custom Management Utilities Using JMX for Oracle WebLogic Server*.

The following topics describe how to view or configure MBeans:

- [Using the System MBean Browser](#)
- [Using the MBeans for a Selected Application](#)

2.5.1 Using the System MBean Browser

You can view the System MBean Browser for many entities, including an Oracle WebLogic Server domain, an Administration Server, a Managed Server, or an application. You can search for an MBean, filter the list of MBeans, and refresh the list of MBeans in the MBean navigation tree.

To view the System MBean Browser specific to a particular Oracle WebLogic Server Managed Server and to configure and use the MBeans:

1. From the target navigation pane, expand the domain.
2. Select the Managed Server.
3. From the WebLogic Server menu, choose **System MBean Browser**.

The System MBean Browser page is displayed.

4. Expand a node in the MBean navigation tree and drill down to the MBean you want to access. Select an MBean instance.

If you do not know the location of an MBean, you can search for the MBean:

- a. Click the Find icon at the top of the MBean navigation tree.
 - b. For **Find**, select **MBean Name**.
You can also select Attributes, Operations, or JMX syntax.
 - c. Enter the name of the MBean and click the arrow.
5. To view the MBean's attributes, select the Attributes tab. Some attributes allow you to change their values. To do so, enter the value in the **Value** column.
 6. To view the available operations, select the Operations tab. To perform an operation, click the operation. The Operations page appears. Enter any applicable values and click **Invoke**.

For more information, see the Fusion Middleware Control online help.

2.5.2 Using the MBeans for a Selected Application

You can view, configure, and use the MBeans for a specific application by taking the steps described in [Section 2.5.1](#), and drilling down to the application. As an alternative, you can navigate to an application's MBeans using the following steps:

1. From the target navigation pane, expand **Application Deployments**.
2. Select the application.
3. From the Application Deployments menu, choose **System MBean Browser**.

The System MBean Browser page is displayed, along with the MBean information for the application.

4. To view the MBean's attributes, select the Attributes tab. Some attributes allow you to change their values. To do so, enter the value in the **Value** column.
5. To view the available operations, select the Operations tab. To perform an operation, click the operation. The Operations page appears. Enter any applicable values and click **Invoke**.

2.6 Managing Components

To manage the Oracle WebLogic Server and Java components, you can use WLST, Oracle WebLogic Server Administration Console, or Fusion Middleware Control.

To manage system components, you can use WLST, or Fusion Middleware Control.

See:

- *Planning an Installation of Oracle Fusion Middleware* and the individual installation guides for information about installing and configuring components
- *Installing and Configuring Oracle WebLogic Server and Coherence* for installing and configuring Oracle WebLogic Server
- The administration guide for each component or suite for more information about managing these components.

2.7 Changing the Administrative User Password

During the Oracle Fusion Middleware installation, you must specify a password for the administration account. Then, you can use this account to log in to Fusion Middleware Control and the Oracle WebLogic Server Administration Console for the first time. You can create additional administrative accounts using the WLST command line or the Oracle WebLogic Server Administration Console.

You can change the password of the administrative user using the command line or the Oracle WebLogic Server Administration Console, as described in the following topics:

- [Changing the Administrative User Password Using the Command Line](#)
- [Changing the Administrative User Password Using the Administration Console](#)

For more information about users, roles, and changing passwords, see "Understanding Users and Roles" in the *Securing Applications with Oracle Platform Security Services*

2.7.1 Changing the Administrative User Password Using the Command Line

To change the administrative user password or other user passwords using the command line, you invoke the `UserPasswordEditorMBean.changeUserPassword` method, which is extended by the security realm's `AuthenticationProvider` MBean.

For more information, see the `changeUserPassword` method in the *MBean Reference for Oracle WebLogic Server*.

2.7.2 Changing the Administrative User Password Using the Administration Console

To change the password of an administrative user using the Oracle WebLogic Server Administration Console:

1. Navigate to the Oracle WebLogic Server Administration Console. (For example, from the home page of the domain in Fusion Middleware Control, select **To configure and managed this WebLogic Domain, use the Oracle WebLogic Server Administration Console**.)
2. From the Domain Structure pane, select **Security Realms**.
The Summary of Security Realms page is displayed.
3. Select a realm, such as **myrealm**.
The Settings for the realm page is displayed.
4. Select the Users and Groups tab, then the Users tab. Select the user.
The Settings for *user* page is displayed.
5. Select the Passwords tab.
6. Enter the new password, then enter it again to confirm it.
7. Click **Save**.

2.8 Configuring Node Manager

Node Manager allows you to perform common operations, such as starting and stopping, for a Managed Server using the Administration Console or Fusion Middleware Control.

This section describes the following topics:

- [Configuring Node Manager to Start Managed Servers](#)
- [Configuring Node Manager to Use the OPSS Keystore Service](#)

2.8.1 Configuring Node Manager to Start Managed Servers

If a Managed Server contains other Oracle Fusion Middleware products, such as Oracle JRF, the Managed Servers environment must be configured to set the correct classpath and parameters. By default, Node Manager is configured when you install Oracle Fusion Middleware.

However, if you do not select automatic configuration, you must provide this environment information through the start scripts, such as `startWebLogic` and `setDomainEnv`, which are located in the following directory:

```
DOMAIN_HOME/bin
```

If the Managed Servers are started by Node Manager (as is the case when the servers are started by the Oracle WebLogic Server Administration Console or Fusion

Middleware Control), Node Manager must be instructed to use these start scripts so that the server environments are correctly configured. Specifically, Node Manager must be started with the property `StartScriptEnabled=true`.

There are several ways to ensure that Node Manager starts with this property enabled. As a convenience, Oracle Fusion Middleware provides the following script, which adds the property `StartScriptEnabled=true` to the `nodemanager.properties` file:

```
(UNIX) #ORACLE_COMMON_HOME/common/bin/setNMProps.sh.
(Windows) #ORACLE_COMMON_HOME\common\bin\setNMProps.cmd
```

For example, on Linux, execute the `setNMProps` script and start Node Manager:

```
ORACLE_COMMON_HOME/common/bin/setNMProps.sh
DOMAIN_HOME/bin/startNodeManager.sh
```

When you start Node Manager, it reads the `nodemanager.properties` file with the `StartScriptEnabled=true` property, and uses the start scripts when it subsequently starts Managed Servers. Note that you need to run the `setNMProps` script only once.

Also note that when the `StartScriptEnable` property is set to true, the Node Manager reads the `startWebLogic` script, which in turns reads the `setDomainEnv` script. As a result, you must make any tuning changes by editing the `setDomainEnv` script. Any changes that are performed using the command line or Administration Console will not be implemented when Node Manager starts the servers. For example, if you use the Administration Console to change the server start arguments, those changes are written to `config.xml`, but the Node Manager ignores these settings and uses those in `setDomainEnv`.

See "Using Node Manager" in the *Administering Node Manager for Oracle WebLogic Server* for other methods of configuring and starting Node Manager.

2.8.2 Configuring Node Manager to Use the OPSS Keystore Service

If you created a domain that included Oracle JRF and you configured Node Manager as "per domain", you can configure Node Manager to use the Oracle Platform Security Services Keystore Service. Take the following steps:

1. Configure the Keystore Service, as described in "Keystore Management with the Keystore Service" in *Securing Applications with Oracle Platform Security Services*.
2. Configure Node Manager by editing the following file:

```
nodemanager.properties
```

In the file, specify the following properties:

```
KeyStores=CustomIdentityAndDemoTrust
CustomIdentityKeyStoreType=KSS
CustomIdentityKeyStoreFileName=kss://system/keystore_name
CustomIdentityKeyStorePassPhrase=#keystore_passphrase
CustomIdentityAlias=#key store alias
CustomIdentityPrivateKeyPassPhrase=#keystore_private_key_passphrase
```

Oracle Platform Security Services Keystore Service is not supported for a "per host" Node Manager. In certain circumstances, however, a "per host" Node Manager will attempt to load the keystore service. To prevent that, you must specify `UseKSSForDemo=false` in the following file:

```
ORACLE_HOME/oracle_common/common/nodemanager/nodemanager.properties#
```

2.9 Basic Tasks for Configuring and Managing Oracle Fusion Middleware

The following provides a summary of the steps you need to take to configure and manage a basic Oracle Fusion Middleware environment after you have installed the software:

1. Configure Oracle WebLogic Server and components, such as Oracle HTTP Server. See *Planning an Installation of Oracle Fusion Middleware*.
2. Configure Node Manager. See [Section 2.8](#).
3. Configure SSL. See [Chapter 6](#).
4. Create and manage metadata repositories, including the MDS Repository. See [Section 13.2](#).
5. Deploy an application. See [Chapter 9](#).
6. Configure load balancing. You can configure load balancing between different components or applications. See the *High Availability Guide*.
7. Back up your environment. See [Chapter 15](#).
8. Monitor your environment and manage log files. See [Chapter 10](#) and [Chapter 11](#).
9. Expand your environment. See [Chapter 18](#).

This guide also describes other tasks that you may need to perform, depending on your Oracle Fusion Middleware environment.

Note: The procedures in this book for the most part assume that you are using the standard installation topology, which consists of a domain that contains an Administration Server and a cluster containing two Managed Servers.

For more information about the standard topology, see "Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology" in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Cross-Component Wiring

This chapter describes service tables and how to wire particular Oracle Fusion Middleware components together.

It contains the following sections:

- [Understanding Service Tables](#)
- [Viewing Service Tables](#)
- [Cross-Component Wiring](#)

3.1 Understanding Service Tables

Service tables provide a way for service providers to publish endpoint information about their services, and clients of these services to query and bind to these services. A service table is a single table in a database schema. There is one row for every endpoint that is published to it. The service table schema is initially created by the Repository Creation Utility.

See "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility* for information about the service table schema.

The local service table is associated with a domain. It contains endpoints that are offered by that domain. For the local service table, the data source name is LocalSvcTblDataSource.

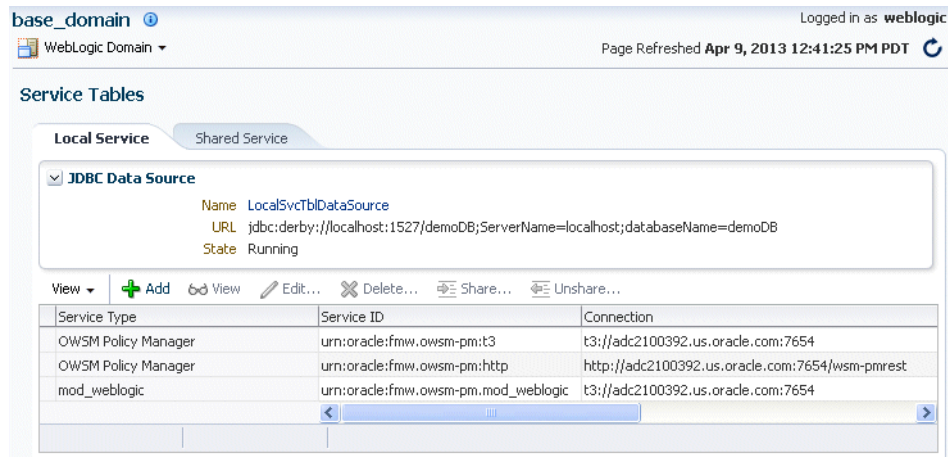
For example, by default, the service table contains endpoint information for Oracle Web Services Manager and Fusion Middleware Control.

3.2 Viewing Service Tables

You can view the service tables using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Cross Component Wiring**, then **Service Tables**.

The Service Tables page is displayed.



2. You can view, edit, or delete the properties of an endpoint by clicking one of the buttons.

Note that a shared service table is used for storing service information between different domains. It is not used in 12.1.2"

3.3 Cross-Component Wiring

When you install and configure Oracle Fusion Middleware, most of the cross-component wiring is automatically performed. However, there may be cases when you want to connect another component or to change the current wiring. For example:

- To connect Oracle HTTP Server to the Administration Server, so that it is connected to Fusion Middleware Control Console and to the Administration Console. See [Section 3.3.1](#).
- To connect Oracle HTTP Server to an Oracle WebLogic Server cluster or server, so that applications can be routed through Oracle HTTP Server to the cluster or server. See [Section 3.3.2](#).
- To connect the Oracle Web Services Manager agent to the Policy Manager. See "Using Cross-Component Wiring for Auto-Discovery of Policy Manager" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3.3.1 Wiring Oracle HTTP Server to the Administration Server

You can connect Oracle HTTP Server to the Administration Server so that you can access Fusion Middleware Control Console and to the Administration Console through the Oracle HTTP Server.

To connect Oracle HTTP Server to the Administration Server:

1. From the navigation pane, expand the domain, then **Web Tier**.
2. Select an Oracle HTTP Server instance, such as ohs1.
The Oracle HTTP Server page is displayed.
3. From the Oracle HTTP Server menu, select **Administration**, then **mod_wl_ohs Configuration**.

The mod_wl_ohs Configuration page is displayed, as shown in the following figure:

ohs_1 | Logged in as weblogic | Page Refreshed May 13, 2013 10:42:27 AM PDT

Oracle HTTP Server | Start Up | Shut Down...

Information
All changes made in this page require a server restart to take effect.

mod_wl_ohs Configuration [Apply] [Revert]

General
The mod_wl_ohs module allows requests to be proxied from an Oracle HTTP Server to Oracle WebLogic Server. Specify parameters common for all URLs directed to the mod_wl_ohs handler, and use the MatchExpression and Location sections to specify any overrides. If both WebLogic Cluster and WebLogic Host are defined, WebLogic Cluster will be used.

WebLogic Cluster [] [Search]
WebLogic Host [] [Search]
WebLogic Port []
 Dynamic Server List ON
Error Page []
WebLogic Temp Directory []
Exclude Path or Mime Type []

Match Expressions
[Add Row] [Remove]
Expression
No data to display

TIP For example MatchExpression can have the following value:
*.jsp WebLogicHost=myHost|WebLogicPort=7001|PathPrepend=/test2

Locations
Click on 'Auto Fill' to get all the valid weblogic server end point locations. This will update any existing location with the same name and add all the new locations to the below table.
[Add Row] [Remove] [Auto Fill]

Location	WebLogic Cluster	WebLogic Host	WebLogic Port	Path Trim	Pat
No Locations Defined					

4. To connect to the Administration Server:
 - For **WebLogic Host**, enter the host name for the Administration Server.
 - For **WebLogic Port**, enter the server port for the Administration Server.

Alternatively, you can click the search icon. Then, select the Administration Server and click **OK**. The fields will be filled in automatically.
5. In the Locations section, click **AutoFill**.
All valid WebLogic Server endpoint locations are displayed.
6. From the table, select /em.
7. To add the Administration Console:
 - a. Click **Add Row**.
 - b. For location, enter /console.
 - c. For **WebLogic Host**, enter the host name for the Administration Server.
 - d. For **Port**, enter the Administration Server port number.
8. Click **Apply**.
9. Shutdown the Oracle HTTP Server instance, then start it again.

3.3.2 Routing Applications Through Oracle HTTP Server to Oracle WebLogic Server

To connect Oracle HTTP Server so that requests are routed through Oracle HTTP Server to Oracle WebLogic Server:

1. From the navigation pane, expand the domain, then **Web Tier**.
2. Select an Oracle HTTP Server instance, such as ohs1.
The Oracle HTTP Server page is displayed.
3. From the Oracle HTTP Server menu, select **Administration**, then **mod_wl_ohs configuration**.

The mod_wl_ohs Configuration page is displayed, as shown in the following figure:

4. To have requests routed to the cluster, for **WebLogic Cluster**, enter the cluster address. Alternatively you can click the search Icon and select the cluster. The fields will be filled automatically.

If you want to have requests routed to a single WebLogic server:

- For **WebLogic Host**, enter the server name.
- For **WebLogic Port**, enter the server port.

Alternatively, you can click the search icon next to **WebLogic Host**. Then, select the server from the list and click **OK**. The fields will be filled in automatically.

Note that if you enter both WebLogic Cluster and WebLogic Host and Port. WebLogic Cluster will be used.

5. In the Locations section, click **AutoFill**.
All valid WebLogic Server endpoint locations are displayed.
6. Select the application from the table.
7. Click **Apply**.
8. Shutdown the Oracle HTTP Server instance, then start it again.

Starting and Stopping Oracle Fusion Middleware

This chapter describes procedures for starting and stopping Oracle Fusion Middleware, including the Administration Server, Managed Servers, and components.

It contains the following sections:

- [Overview of Starting and Stopping Procedures](#)
- [Starting and Stopping Oracle WebLogic Server Administration and Managed Servers](#)
- [Starting and Stopping Components](#)
- [Starting and Stopping Fusion Middleware Control](#)
- [Starting and Stopping Applications](#)
- [Starting and Stopping Your Oracle Fusion Middleware Environment](#)
- [Starting and Stopping: Special Topics](#)

4.1 Overview of Starting and Stopping Procedures

Oracle Fusion Middleware is a flexible product that you can start and stop in different ways, depending on your requirements. In most situations, you can use Fusion Middleware Control, Oracle WebLogic Server Administration Console, or the WLST commands to start or stop Oracle Fusion Middleware components.

These tools are completely compatible and, in most cases, can be used interchangeably. For example, you can start a J2EE component using WLST and stop it using Fusion Middleware Control.

4.2 Starting and Stopping Oracle WebLogic Server Administration and Managed Servers

You can start Oracle WebLogic Server Administration Servers using the WLST command line. You can start and stop Managed Servers using scripts, the WLST command line, the WebLogic Server Administration Console, or Fusion Middleware Control. The following sections describe how to start and stop WebLogic Servers using the WLST command line, Fusion Middleware Control, or both:

- [Starting and Stopping Administration Server](#)
- [Starting and Stopping Node Manager](#)
- [Starting and Stopping Managed Servers](#)

- [Enabling Servers to Start Without Supplying Credentials](#)
- [Setting Up Oracle WebLogic Server as a Windows Service](#)

4.2.1 Starting and Stopping Administration Server

You can start and stop the Oracle WebLogic Server Administration Server using the WLST command line or a script. When you start or stop the Administration Server, you also start or stop the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

For example, to start an Administration Server, use the following script:

```
DOMAIN_HOME/bin/startWebLogic.sh
```

To stop an Administration Server, use the following script:

```
DOMAIN_HOME/bin/stopWebLogic.sh#  
#####username#password#[admin_url]
```

4.2.2 Starting and Stopping Node Manager

By default, Node Manager is configured when you configure Oracle Fusion Middleware. If Node Manager is not configured, it is very important to change the Node Manager property `StartScriptEnabled` to `True`. If this property is set to `False` you will encounter errors or problems when starting Managed Servers configured for use by Oracle Fusion Middleware components. See [Section 2.8.1](#) for more information.

You can start Node Manager using the WLST command line or a script.

To start Node Manager, use the following script:

```
(UNIX) #DOMAIN_HOME/bin/startNodeManager.sh  
(Windows) #DOMAIN_HOME\bin\startNodeManager.cmd
```

To stop Node Manager, close the command shell in which it is running.

Alternatively, after having set the `nodemanager.properties` attribute `QuitEnabled` to `true` (the default is `false`), you can use WLST to connect to Node Manager and shut it down. For more information, see `stopNodeManager` in the *WLST Command Reference for WebLogic Server*.

4.2.3 Starting and Stopping Managed Servers

You can start and stop Managed Servers using Fusion Middleware Control or WLST commands and scripts, as described in the following topics:

- [Starting and Stopping Managed Servers Using Fusion Middleware Control](#)
- [Starting and Stopping Managed Servers Using Scripts](#)

4.2.3.1 Starting and Stopping Managed Servers Using Fusion Middleware Control

Fusion Middleware Control and the Oracle WebLogic Server Administration Console use Node Manager to start Managed Servers. If you are starting a Managed Server that does not contain Oracle Fusion Middleware products other than Oracle WebLogic Server, you can start the servers using the procedure in this section.

However, if the Managed Server contains other Oracle Fusion Middleware products, such as Oracle JRF, you must first configure Node Manager, as described in [Section 2.8.1](#).

To start or stop a WebLogic Server Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the Managed Server.
3. From the WebLogic Server menu, choose **Control**, then **Start Up** or **Shut Down**.

Alternatively, you can right-click the server, then choose **Control**, then **Start Up** or **Shut Down**.

4.2.3.2 Starting and Stopping Managed Servers Using Scripts

You can use a script or WLST to start and stop a WebLogic Server Managed Server.

For example, to start a WebLogic Server Managed Server, use the following script:

```
(UNIX)#DOMAIN_HOME/bin/startManagedWebLogic.sh
#####managed_server_name#admin_url#
(Windows)#DOMAIN_HOME\bin\startManagedWebLogic.cmd
#####managed_server_name#admin_url
```

When prompted, enter your user name and password.

To stop a WebLogic Server Managed Server, use the following script:

```
(UNIX)#DOMAIN_HOME/bin/stopManagedWebLogic.sh
#####managed_server_name#admin_url#username#password#
(Windows)#DOMAIN_HOME\bin\stopManagedWebLogic.cmd#
#####managed_server_name#admin_url#username#password
```

For information about using WLST to start and stop Managed Servers, see "Managing the Server Life Cycle" in *Understanding the WebLogic Scripting Tool*.

4.2.4 Enabling Servers to Start Without Supplying Credentials

You can enable the Administration Server and Managed Servers to start without prompting you for the administrator user name and password.

1. For the Administration Server, create a boot.properties file:

- a. Create the following directory:

```
DOMAIN_HOME/servers/AdminServer/security
```

- b. Use a text editor to create a file called boot.properties in the security directory created in the previous step, and enter the following lines in the file:

```
username=adminuser
password=password
```

2. For each Managed Server:

- a. Create the following directory:

```
DOMAIN_HOME/servers/server_name/security
```

- b. Copy the boot.properties file you created for the Administration Server to the security directory you created in the previous step.

3. Restart the Administration Server and Managed Servers, as described in [Section 4.2.1](#) and [Section 4.2.3](#).

Note: When you start the Administration Server or Managed Server, the user name and password entries in the file are encrypted.

For security reasons, minimize the time the entries in the file are left unencrypted. After you edit the file, start the server as soon as possible in order for the entries to be encrypted.

For more information, see "Boot Identity Files" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

4.2.5 Setting Up Oracle WebLogic Server as a Windows Service

If you want a WebLogic Server instance to start automatically when you boot a Windows host computer, you can set up the server as a Windows service. For complete information, see "Setting Up a WebLogic Server Instance as a Windows Service" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

However, that chapter describes the process for a standalone Oracle WebLogic Server installation. When Oracle WebLogic Server is part of an Oracle Fusion Middleware environment, you must set the environment to include references to `ORACLE_COMMON`. To do that, the script that you create is slightly different from that in "Example Script for Setting Up a Server as a Windows Service". The following shows the correct script:

```
echo#off
SETLOCAL
set#DOMAIN_NAME=myWLSdomain
set#USERDOMAIN_HOME=d:\Oracle\config\domains\myWLSdomain
set#SERVER_NAME=myWLSserver
set#PRODUCTION_MODE=true
set
JAVA_OPTIONS=-Dweblogic.Stdout="d:\Oracle\config\domains\myWLSdomain\
stdout.txt" #-Dweblogic.Stderr="d:\Oracle\config\domains\myWLSdomain\stderr.txt"
set#ADMIN_URL=http://adminserver:7501
set#MEM_ARGS=-Xms40m#-Xmx250m
call#%USERDOMAIN_HOME%\bin\setDomainEnv.cmd
call#"d:\Oracle_home\wlserver\server\bin\installSvc.cmd"
ENDLOCAL
```

4.3 Starting and Stopping Components

You can start and stop components using the command line, the WebLogic Server Administration Console, or Fusion Middleware Control, depending upon the component. The following topics describe how to start and stop components using Fusion Middleware Control and the command line:

- [Starting and Stopping Components Using Fusion Middleware Control](#)
- [Starting and Stopping Components Using the Command Line](#)

4.3.1 Starting and Stopping Components Using Fusion Middleware Control

To start or stop a component:

1. From the navigation pane, navigate to the component.
2. Select the component, such as **OHS**.
3. From the dynamic target menu, choose **Control**, then **Start Up** or **Shut Down**.

4.3.2 Starting and Stopping Components Using the Command Line

If a component is a Java component, you use WLST commands to start and stop the component. If a component is a system component, you can use scripts to call WLST commands to start and stop the components, as described in the following topics:

- [Starting and Stopping Java Components](#)
- [Starting and Stopping System Components](#)

4.3.2.1 Starting and Stopping Java Components

To start and stop Java components, use the WLST `startApplication` and `stopApplication` commands:

```
startApplication(appName,#[options])
stopApplication(appName,#[options])
```

For example, to start Oracle Web Services Manager Policy Manager, use the following command:

```
startApplication("wsm-pm")
```

4.3.2.2 Starting and Stopping System Components

If a component is a system component, you can use scripts to call WLST commands to start and stop the components or you can use WLST commands:

- To start and stop system components using scripts, use the `startComponent` and `stopComponent` scripts. You can use this method for Oracle HTTP Server in a standalone domain or a WebLogic Server domain.

The scripts are located in

```
(UNIX) #DOMAIN_HOME/bin#
(Windows) #DOMAIN_HOME\bin#
```

To start or stop a component using these scripts, use the following syntax:

```
./startComponent.sh#component_name#[storeUserConfig]#[showErrorStack]
./stopComponent.sh#component_name#[storeUserConfig]#[showErrorStack]
```

In the syntax:

- *component_name*: The name of the component instance, such as ohs1.
- *storeUserConfig*: When specified, the script will prompt you for the user name and password. Then, it will ask you if you want to store the user configuration in a properties file. If you specify *y*, it creates a user configuration file and an associated key file. The user configuration file contains an encrypted user name and password. The key file contains a secret key that is used to encrypt and decrypt the user name and password. The following shows the names and location of the properties files:

```
user_home/.wlst/nm-key-domain_name.props
user_home/.wlst/nm-cfg-domain_name.props
```

After you have stored the information in the properties file, when you run the scripts subsequently, you will not be prompted for a user name and password.

- *showErrorStack*: Provides more detailed error information, including all of the messages in the error stack. Specify this option if you need to determine the cause of errors.

For example, to start an Oracle HTTP Server instance called ohs1:

```
./startComponent.sh#ohs1
```

You can also use these scripts to start and stop system components remotely. In that case, the scripts read the configuration to determine the location of the component.

- To start system components using WLST commands, you can use one of the following methods:
 - The nmstart command. You can use this method for Oracle HTTP Server in a standalone domain or a WebLogic Server domain.

For example, to start the Oracle HTTP Server component OHS1, use the following WLST commands:

```
nmConnect (domainName='domain_name', #username='username', #
password='password')
nmstart (serverName='OHS1', #serverType='OHS')
```

- The WLST start command. You can use this method for Oracle HTTP Server in a standalone domain.

For example, to start the Oracle HTTP Server component OHS1, use the following WLST commands:

```
connect ('username', 'password', 'hostname:port')
start('OHS1')
```

- To stop system components using WLST commands, use the WLST nmkill command.

For example, to kill the Oracle HTTP Server component OHS1, use the following WLST commands:

```
nmKill (serverName='ohs1', #serverType='OHS')
```

To decide which method to use, note the following:

- If you are using a WLST script, use the WLST commands.
- To quickly start and stop system components interactively, use the scripts.
- To start and stop system components remotely, use the scripts.

4.4 Starting and Stopping Fusion Middleware Control

If Fusion Middleware Control is configured for a domain, it is automatically started or stopped when you start or stop an Oracle WebLogic Server Administration Server, as described in [Section 4.2.1](#).

4.5 Starting and Stopping Applications

You can start and stop applications using Fusion Middleware Control, the WebLogic Server Administration Console, or the WLST command line. The following topics describe how to start and stop applications using Fusion Middleware Control and the command line:

- [Starting and Stopping Java EE Applications Using Fusion Middleware Control](#)
- [Starting and Stopping Java EE Applications Using WLST](#)

4.5.1 Starting and Stopping Java EE Applications Using Fusion Middleware Control

To start or stop a Java EE application using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments**.
2. Select the application.
3. From the Application Deployment menu, choose **Control**, then **Start Up** or **Shut Down**.

4.5.2 Starting and Stopping Java EE Applications Using WLST

To start or stop a Java EE application with the WLST command line, use the following commands:

```
startApplication(appName,#[options])
stopApplication(appName,#[options])
```

The application must be fully configured and available in the domain. The `startApplication` command returns a `WLSTProgress` object that you can access to check the status of the command. In the event of an error, the command returns a `WLSTException`. For more information about the `WLSTProgress` object, see "WLSTProgress Object" in *Understanding the WebLogic Scripting Tool*.

4.6 Starting and Stopping Your Oracle Fusion Middleware Environment

This section provides procedures for starting and stopping an Oracle Fusion Middleware environment. An environment can consist of an Oracle WebLogic Server domain, an Administration Server, multiple Managed Servers, Java components, system components, including Identity Management components, and a database used as a repository for metadata. The components may be dependent on each other. Therefore, it is important to start and stop them in a particular order.

4.6.1 Starting an Oracle Fusion Middleware Environment

To start an Oracle Fusion Middleware environment:

1. Start the database that hosts the metadata schemas. The following steps illustrate one method for starting the database.
 - a. Navigate to the location of the database. For example, the database may reside on a different host than Oracle Fusion Middleware.
 - b. Set the `ORACLE_HOME` environment variable to the Oracle home for the database.
 - c. Set the `ORACLE_SID` environment variable to the SID for the database (default is `orcl`.)
 - d. Start the Net Listener:

```
ORACLE_HOME/bin/lsnrctl#start
```

- e. Start the database instance:

```
ORACLE_HOME/bin/sqlplus#/nolog
SQL>#connect#SYS#as#SYSDBA
SQL>#startup
SQL>#quit
```

For more information about starting an Oracle Database, see the *Oracle Database Administrator's Guide*.

2. Start the Oracle WebLogic Server Administration Server as described in [Section 4.2.1](#).
3. Start Node Manager as described in [Section 4.2.2](#).
4. Start any Oracle Identity Management components, such as Oracle Internet Directory, which form part of your environment.
5. Start the Oracle WebLogic Server Managed Servers as described in [Section 4.2.3.2](#).

Note: The start up of a Managed Server will typically start up the applications which are deployed to it. Therefore, it should not be necessary to manually start applications after the Managed Server startup.

6. Start all other system components, such as Oracle HTTP Server:

```
(UNIX) #DOMAIN_HOME/bin/startComponent.sh#component_name
(Windows) #DOMAIN_HOME\bin\startComponent.cmd#component_name
```

4.6.2 Stopping an Oracle Fusion Middleware Environment

To stop an Oracle Fusion Middleware environment:

1. Stop system components, such as Oracle HTTP Server. You can stop them in any order:

```
(UNIX) #DOMAIN_HOME/bin/stopComponent.sh#component_name#
(Windows) #DOMAIN_HOME\bin\stopComponent.cmd#component_name
```

2. Stop the Oracle WebLogic Server Managed Servers, as described in [Section 4.2](#). Any applications deployed to the server are also stopped.
3. Stop any 11g Oracle Identity Management components, such as Oracle Internet Directory, which form part of your environment.
4. Stop the Administration Server as described in [Section 4.2.1](#).
5. Stop Node Manager as described in [Section 4.2.2](#).
6. Start the database that hosts the metadata schemas. The following steps illustrate one method for stopping the database.
 - a. Navigate to the location of the database. For example, the database may reside on a different host than Oracle Fusion Middleware.
 - b. Set the ORACLE_HOME environment variable to the Oracle home for the database.
 - c. Set the ORACLE_SID environment variable to the SID for the database (default is orcl).
 - d. Stop the database instance:


```
ORACLE_HOME/bin/sqlplus#/nolog
SQL>#connect#SYS#as#SYSDBA
SQL>#shutdown
SQL>#quit
```
 - e. Stop the Net Listener:

```
ORACLE_HOME/bin/lsnrctl#stop
```

For more information about stopping an Oracle Database, see the *Oracle Database Administrator's Guide*.

4.7 Starting and Stopping: Special Topics

This section contains the following special topics about starting and stopping Oracle Fusion Middleware:

- [Starting and Stopping in High Availability Environments](#)
- [Forcing a Shutdown of Oracle Database](#)

4.7.1 Starting and Stopping in High Availability Environments

There are special considerations and procedures for starting and stopping High Availability environments, such as:

- Oracle Fusion Middleware Cold Failover Cluster
- Oracle Application Server Disaster Recovery

See the *High Availability Guide* for information about starting and stopping in high-availability environments.

4.7.2 Forcing a Shutdown of Oracle Database

If you find that the Oracle Database instance is taking a long time to shut down, you can use the following commands to force an immediate shutdown:

```
ORACLE_HOME/bin/sqlplus#/nolog
SQL>#connect#SYS#as#SYSDBA
SQL>#SHUTDOWN#IMMEDIATE;
```

An immediate database shutdown proceeds with the following conditions:

- No new connections are allowed, nor are new transactions allowed to be started, after the statement is issued.
- Any uncommitted transactions are rolled back. (If long uncommitted transactions exist, this method of shutdown might not complete quickly, despite its name.)
- Oracle does not wait for users currently connected to the database to disconnect. Oracle implicitly rolls back active transactions and disconnects all connected users.

The next startup of the database will not require any instance recovery procedures.

For more information about shutting down an Oracle Database, *Oracle Database Administrator's Guide* in the Oracle Database documentation library

Managing Ports

This chapter describes how to view and change Oracle Fusion Middleware port numbers, such as port numbers used by Oracle WebLogic Server or Oracle HTTP Server.

It contains the following sections:

- [About Managing Ports](#)
- [Viewing Port Numbers](#)
- [Changing the Port Numbers Used by Oracle Fusion Middleware](#)

5.1 About Managing Ports

Many Oracle Fusion Middleware components and services use ports. Most port numbers are assigned during domain creation. As an administrator, it is important to know the port numbers used by these services, and to ensure that the same port number is not used by two services on your host.

For some ports, you can specify a port number assignment during domain creation.

See Also: [Appendix D](#) for a list of port numbers. Refer to the installation guide for directions on overriding port assignments during installation.

5.2 Viewing Port Numbers

You can view the port numbers currently in use with the command line or Fusion Middleware Control, as described in the following topics:

- [Viewing Port Numbers Using the Command Line](#)
- [Viewing Port Numbers Using Fusion Middleware Control](#)

5.2.1 Viewing Port Numbers Using the Command Line

To view the port numbers for Oracle WebLogic Server, you can use the WLST `get` command, with an attribute. For example, to get the Administration Port, use the following command:

```
wls:/WLS_domain/serverConfig>#get('AdministrationPort')
9002
```

5.2.2 Viewing Port Numbers Using Fusion Middleware Control

You can view the port numbers of the domain, the Administration Server, Managed Servers, or components, such as Oracle HTTP Server, using Fusion Middleware Control.

For example, to view the ports of a domain:

1. From the navigation pane, select the domain.
2. From the WebLogic Domain menu, choose **Monitoring**, then **Port Usage**.

The Port Usage page is displayed, as shown in the following figure:

Port in Use	IP Address	Component	Channel	Protocol
7001	127.0.0.1	AdminServer	Default[http]	http
7001	127.0.0.1	AdminServer	Default[http][4]	http
7001	127.0.0.1	AdminServer	Default[ldap]	ldap
7004	127.0.0.1	wls_server_2	Default[CLUSTER-BR...]	CLUSTER-BROADCAST
7003	127.0.0.1	wls_server_1	Default[t3]	t3
7001	127.0.0.1	AdminServer	Default[snmp][4]	snmp
7001	127.0.0.1	AdminServer	Default[ldap][1]	ldap
7001	127.0.0.1	AdminServer	Default[http][2]	http
7001	0:0:0:0:0:0:1	AdminServer	Default[http][3]	http
7001	127.0.0.1	AdminServer	Default[snmp]	snmp
7001	127.0.0.1	AdminServer	Default[iiop][1]	iiop
7004	127.0.0.1	wls_server_2	Default[snmp]	snmp
7001	0:0:0:0:0:0:1	AdminServer	Default[snmp][3]	snmp
7003	127.0.0.1	wls_server_1	Default[CLUSTER-BR...]	CLUSTER-BROADCAST
7001	127.0.0.1	AdminServer	Default[t3]	t3
7001	127.0.0.1	AdminServer	Default[t3][1]	t3
7001	127.0.0.1	AdminServer	Default[ldap][2]	ldap
7001	0:0:0:0:0:0:1	AdminServer	Default[iiop][3]	iiop

Optionally, you can filter the ports shown by selecting a Managed Server from **Show**.

The Port Usage detail table shows the ports that are in use, the IP Address, the component, the channel, and the protocol.

You can also view similar pages for the Administration Server, Managed Servers, and components, such as Oracle HTTP Server, by navigating to the target and choosing **Port Usage** from the target's menu.

5.3 Changing the Port Numbers Used by Oracle Fusion Middleware

You can change the port numbers for some Oracle Fusion Middleware components, using Fusion Middleware Control, Oracle WebLogic Server Administration Console, or the command line.

Note: You can change a port number to any number you want, if it is an unused port. You do not have to use a port in the allotted port range for the component. See [Appendix D](#) for information on allotted port ranges.

This section provides the following topics:

- [Changing the Oracle WebLogic Server Listen Ports](#)
- [Changing the Oracle HTTP Server Listen Ports](#)

- [Changing the Oracle Database Net Listener Port](#)

For information about changing other ports, see:

- "Overview of Node Manager Configuration" in the *Administering Node Manager for Oracle WebLogic Server* for information about changing the Node Manager port.

5.3.1 Changing the Oracle WebLogic Server Listen Ports

You can change the non-SSL (HTTP) listen port and the SSL (HTTPS) listen port for a WebLogic Server Administration Server or a Managed Server using the Oracle WebLogic Server Administration Console or WLST, as described in the following topics:

- [Changing the Oracle WebLogic Server Listen Ports Using the Administration Console](#)
- [Changing the Oracle WebLogic Server Listen Ports Using WLST](#)

See *Administering Server Environments for Oracle WebLogic Server* for more information about changing Oracle WebLogic Server ports.

5.3.1.1 Changing the Oracle WebLogic Server Listen Ports Using the Administration Console

To change the non-SSL (HTTP) listen port and the SSL (HTTPS) listen port for a WebLogic Server Administration Server or a Managed Server using the Oracle WebLogic Server Administration Console:

1. Navigate to the server.
The Settings for *server_name* page is displayed.
2. Select the **Configuration** tab. On the General tab, change the number of the **Listen Port** or **SSL Listen Port**.
3. If the server is running, restart the server.
4. If other components rely on the Oracle WebLogic Server listen ports, you must reconfigure those components.

5.3.1.2 Changing the Oracle WebLogic Server Listen Ports Using WLST

To change the non-SSL (HTTP) listen port and the SSL (HTTPS) listen port for a WebLogic Server Administration Server or a Managed Server using the WLST command line. You must run the commands in offline mode; that is, you must not be connected to a server.

For example to change the Administration Server HTTP listen port to port 8001, use the following WLST commands:

```
readDomain("oracle/config/domains/domain_name")
cd("servers/AdminServer")
cmo.setListenPort(8001)
updateDomain()
```

5.3.2 Changing the Oracle HTTP Server Listen Ports

To change the Oracle HTTP Server Listen ports (non-SSL or SSL), there are often dependencies that must also be set.

The following topics describe how to modify the Oracle HTTP Server HTTP or HTTPS Listen port:

- [Enabling Oracle HTTP Server to Run as Root for Ports Set to Less Than 1024 \(UNIX Only\)](#)
- [Changing the Oracle HTTP Server Non-SSL Listen Port in a WebLogic Server Domain](#)
- [Changing the Oracle HTTP Server SSL Listen Port in a WebLogic Server Domain](#)
- [Changing the Oracle HTTP Server Listen Ports in a Standalone Domain](#)

5.3.2.1 Enabling Oracle HTTP Server to Run as Root for Ports Set to Less Than 1024 (UNIX Only)

On a UNIX system, if you are changing the Listen port to a number less than 1024, perform these steps before you change the Oracle HTTP Server Listen port.

By default, Oracle HTTP Server runs as a non-root user (the user that installed Oracle Fusion Middleware). On UNIX systems, if you change the Oracle HTTP Server Listen port number to a value less than 1024, you must enable Oracle HTTP Server to run as root, as follows:

1. Log in as root.
2. Run the following commands in the Oracle home:

```
cd#ORACLE_HOME/ohs/bin
chown#root#.apachectl
chmod#6750#.apachectl
```

5.3.2.2 Changing the Oracle HTTP Server Non-SSL Listen Port in a WebLogic Server Domain

To change the Oracle HTTP Server non-SSL (HTTP) Listen port, follow the procedures in the following tasks. Note that, on a UNIX system, if you are changing the Listen port to a number less than 1024, you must first perform the steps in [Section 5.3.2.1](#).

To change the Oracle HTTP Server Listen port using Fusion Middleware Control:

1. From the navigation pane, expand **Web Tier**, then select the Oracle HTTP Server instance.
2. From the Oracle HTTP Server menu, choose **Administration**, then **Ports Configuration**.
3. Select the Listen port that uses the HTTP protocol, then click **Edit**.
4. Change the port number, then click **OK**.
5. Restart Oracle HTTP Server. (From the Oracle HTTP Server menu, choose **Control**, then **Restart**.)

5.3.2.3 Changing the Oracle HTTP Server SSL Listen Port in a WebLogic Server Domain

To change the Oracle HTTP Server SSL (HTTPS) Listen port, follow the procedures in the following tasks. Note that, on a UNIX system, if you are changing the Listen port to a number less than 1024, you must perform the steps in [Section 5.3.2.1](#).

To change the Oracle HTTP Server SSL Listen port using Fusion Middleware Control:

1. From the navigation pane, expand **Web Tier**, then select the Oracle HTTP Server instance.

2. From the Oracle HTTP Server menu, choose **Administration**, then **Ports Configuration**.
3. Select the Listen port that uses the HTTPS protocol, then click **Edit**.
4. Change the port number, then click **OK**.
5. Restart Oracle HTTP Server. (From the Oracle HTTP Server menu, choose **Control**, then **Restart**.)

5.3.2.4 Changing the Oracle HTTP Server Listen Ports in a Standalone Domain

To change the Oracle HTTP Server non-SSL and SSL Listen Ports, modify the following files:

```
DOMAIN_HOME/config/fmwconfig/components/OHS/instances/component_name/httpd.conf
DOMAIN_HOME/config/fmwconfig/components/OHS/instances/component_name/admin.conf
DOMAIN_HOME/config/fmwconfig/components/OHS/instances/component_name/ssl.conf
```

5.3.3 Changing the Oracle Database Net Listener Port

If your environment includes an Oracle Database that functions as a metadata repository, and you want to change the listener port number for that database, perform the procedure in this section.

First, determine if it is necessary to change the listener port number. If you are concerned that you have another database on your host using the same port, both databases can possibly use the same port.

Note that multiple Oracle Database 10g and Oracle Database 11g databases can share the same Oracle Net listener port. If you are using an Oracle Database as a metadata repository on the same host that contains another Oracle Database 10g or Oracle Database 11g database, they can all use port 1521. There is no need to change the listener port number.

Note: To run two listeners that use the same key value on one host, refer to [Section 5.3.3.1, "Changing the KEY Value for an IPC Listener"](#)

The procedure consists of the following tasks:

- [Task 1, "Stop Components"](#)
- [Task 2, "Change the Metadata Repository for Oracle Net Listener Port"](#)
- [Task 3, "Change the System Data Source"](#)

Task 1 Stop Components

Stop all components that use the Metadata Repository. See [Chapter 4](#) for instructions.

Task 2 Change the Metadata Repository for Oracle Net Listener Port

On the metadata repository host:

1. Ensure that the ORACLE_HOME and ORACLE_SID environment variables are set.
2. Stop the metadata repository listener:

```
lsnrctl#stop
```

3. Edit the `listener.ora` file, which is located at:

```
(UNIX) #ORACLE_HOME/network/admin/listener.ora
(Windows) #ORACLE_HOME\network\admin\listener.ora
```

Under the `LISTENER` entry, update the value for `PORT`. Save the file.

4. Edit the `tnsnames.ora` file. The default location is:

```
(UNIX) #ORACLE_HOME/network/admin/tnsnames.ora
(Windows) #ORACLE_HOME\network\admin\tnsnames.ora
```

Make the following changes to the file:

- a. Update the `PORT` value in each entry that applies to MDS Repository.
- b. Add an entry similar to the following:

```
newnetport#=
## (DESCRIPTION#=
##### (ADDRESS#=# (PROTOCOL#=#tcp) # (HOST#=#hostname) # (PORT#=#port)))
```

In the example, *hostname* is the fully qualified host name and *port* is the new port number.

5. Start the metadata repository listener:

```
lsnrctl#start
```

6. Using SQL*Plus, log in to the metadata repository as the `SYSTEM` user with `SYSDBA` privileges and run the following command:

```
SQL> #ALTER#SYSTEM#SET#local_listener='newnetport'#scope=spfile;
```

7. Using SQL*Plus, restart the metadata repository:

```
SQL> #SHUTDOWN
SQL> #STARTUP
```

Task 3 Change the System Data Source

Change the system data source to use the new port number for the metadata repository. To do so, you can use Fusion Middleware Control:

1. In the Change Center, click **Lock & Edit**.
2. In the navigation pane, expand select the domain.
The WebLogic Domain page is displayed.
3. From the WebLogic Domain menu, select **JDBC Data Sources**.
The Summary of JDBC Data Sources page is displayed.
4. Select the data source you want to change.
The JDBC Data Source page is displayed.
5. Select the Connection Pool tab.
6. To change the database port, modify the **Database URL** field. For example:
`jdbc:oracle:thin:@hostname.domainname.com:1522/orcl`
7. Click **Save**.

8. Restart the servers that use this data source. (Click the Targets tab to see the servers that use this data source.)

5.3.3.1 Changing the KEY Value for an IPC Listener

It is not possible to run two listeners at the same time that are configured to use the same KEY value in their IPC protocol address. By default, the metadata repository listener has its IPC KEY value set to EXTPROC. Hence, if your computer has another IPC listener that uses the EXTPROC key, you should configure the metadata repository listener to use some other key value such as EXTPROC1.

To change the KEY value of an IPC listener:

1. Stop the listener (ensure that your ORACLE_HOME environment variable is set first):

```
lsnrctl#stop
```

2. Edit the listener.ora and tnsnames.ora files. In each file, find the following line:

```
(ADDRESS#=# (PROTOCOL#=#IPC) (KEY#=#EXTPROC))
```

Change it to the following:

```
(ADDRESS#=# (PROTOCOL#=#IPC) (KEY#=#EXTPROC1))
```

3. Restart the listener:

```
lsnrctl#start
```


Part III

Secure Sockets Layer

This part describes how to secure communications between Oracle Fusion Middleware components using the Secure Sockets Layer (SSL) and how to use Oracle Fusion Middleware security features to administer keystores, wallets, and certificates.

[Part III](#) contains the following chapters:

- [Chapter 6, "Configuring SSL in Oracle Fusion Middleware"](#)
- [Chapter 7, "Managing Keystores, Wallets, and Certificates"](#)

Configuring SSL in Oracle Fusion Middleware

You can secure communications between Oracle Fusion Middleware components using SSL, which is an industry standard for securing communications. Oracle Fusion Middleware supports TLS version 1.

This chapter provides an overview of SSL and how you can use it with Oracle Fusion Middleware components and applications. It contains these topics:

- [How SSL Works](#)
- [About SSL in Oracle Fusion Middleware](#)
- [Configuring SSL for Configuration Tools](#)
- [Configuring SSL for the Web Tier](#)
- [Configuring SSL for the Middle Tier](#)
- [Configuring SSL for the Data Tier](#)
- [Advanced SSL Scenarios](#)
- [Best Practices for SSL](#)
- [WLST Reference for SSL](#)

Note: Where SSL connections are configured within Oracle WebLogic Server, this chapter provides references to the relevant Oracle WebLogic Server documentation rather than duplicating the instructions here.

6.1 How SSL Works

This section introduces basic SSL concepts. It contains these topics:

- [What SSL Provides](#)
- [About Private and Public Key Cryptography](#)
- [Keystores and Wallets](#)
- [How SSL Sessions Are Conducted](#)

6.1.1 What SSL Provides

SSL secures communication by providing message encryption, integrity, and authentication. The SSL standard allows the involved components (such as browsers and HTTP servers) to negotiate which encryption, authentication, and integrity mechanisms to use.

- Encryption provides confidentiality by allowing only the intended recipient to read the message. SSL can use different encryption algorithms to encrypt messages. During the SSL handshake that occurs at the start of each SSL session, the client and the server negotiate which algorithm to use. Examples of encryption algorithms supported by SSL include AES, RC4, and 3DES.
- Integrity ensures that a message sent by a client is received intact by the server, untampered. To ensure message integrity, the client hashes the message into a digest using a hash function and sends this message digest to the server. The server also hashes the message into a digest and compares the digests. Because SSL uses hash functions that make it computationally infeasible to produce the same digest from two different messages, the server can tell that if the digests do not match, someone had tampered with the message. An example of a hash function supported by SSL is SHA1.
- Authentication enables the server and client to check that the other party is who it claims to be. When a client initiates an SSL session, the server typically sends its certificate to the client. Certificates are digital identities that are issued by trusted certificate authorities, such as Verisign. [Chapter 7, "Managing Keystores, Wallets, and Certificates"](#) describes certificates in more detail.

The client verifies that the server is authentic and not an imposter by validating the certificate chain in the server certificate. The server certificate is guaranteed by the certificate authority (CA) who signed the server certificate.

The server can also require the client to have a certificate, if the server needs to authenticate the identity of the client.

6.1.2 About Private and Public Key Cryptography

To provide message integrity, authentication, and encryption, SSL uses both private and public key cryptography.

Secret Key Cryptography

Symmetric key cryptography requires a single, secret key shared by two or more parties to secure communication. This key is used to encrypt and decrypt secure messages sent between the parties. It requires prior and secure distribution of the key to each party. The problem with this method is that it is difficult to securely transmit and store the key.

In SSL, each party calculates the secret key individually using random values known to each side. The parties then send encrypted messages using the secret key.

Public Key Cryptography

Public key cryptography solves this problem by employing public and private key pairs and a secure method for key distribution. The freely available public key is used to encrypt messages that can *only* be decrypted by the holder of the associated private key. Together with other security credentials, private key is securely stored in an encrypted container such as an Oracle wallet.

Public key algorithms can guarantee the secrecy of a message. However, they do not necessarily guarantee secure communication because they do not verify the identities of the communicating parties. To establish secure communication, it is important to verify that the public key used to encrypt a message does in fact belong to the target recipient. Otherwise, a third party can potentially eavesdrop on the communication and intercept public key requests, substituting its own public key for a legitimate key (the man-in-the-middle attack).

To avoid such an attack, it is necessary to verify the owner of the public key with a process called authentication. trusted by both of the communicating parties, a third party known as a certificate authority (CA) can accomplish the authentication process.

The CA issues public key certificates that contain an entity's name, public key, and certain other security credentials. Such credentials typically include the CA name, the CA signature, and the certificate effective dates (From Date, To Date).

The CA uses its private key to encrypt a message, while the public key is used to decrypt it, thus verifying that the message was encrypted by the CA. The CA public key is well known, and does not have to be authenticated each time it is accessed. Such CA public keys are stored in wallets.

6.1.3 Keystores and Wallets

In Oracle Fusion Middleware, some components use the Oracle Wallet as their storage mechanism. An Oracle wallet is a container that stores your credentials, such as certificates, trusted certificates, certificate requests, and private keys. You can store Oracle wallets on the file system or in LDAP directories such as Oracle Internet Directory. Oracle wallets can be auto-login or password-protected wallets.

Oracle HTTP Server uses Oracle wallet. Configuring SSL for Oracle HTTP Server thus requires setting up and using Oracle wallets.

Other components use a JKS keystore or KSS keystore to store keys and certificates, and configuring SSL for these components requires setting up and using the appropriate keystores.

For more information about configuring keystores and wallets, see:

- [Section 6.2, "About SSL in Oracle Fusion Middleware"](#) for a fuller description of keystore and wallet usage in Oracle Fusion Middleware
- [Chapter 7, "Managing Keystores, Wallets, and Certificates"](#) for a discussion of these terms, and administration details

6.1.4 How SSL Sessions Are Conducted

The SSL protocol has two phases: the handshake phase and the data transfer phase. The handshake phase authenticates the server and optionally the client, and establishes the cryptographic keys that will be used to protect the data to be transmitted in the data transfer phase.

When a client requests an SSL connection to a server, the client and server first exchange messages in the handshake phase. (A common scenario is a browser requesting a page using the `https://` instead of `http://` protocol from a server. The HTTPS protocol indicates the usage of SSL with HTTP.)

[Figure 6–1](#) shows the handshake messages for a typical SSL connection between a Web server and a browser. The following steps are shown in the figure:

1. The client sends a Hello message to the server.
The message includes a list of algorithms supported by the client and a random number that will be used to generate the keys.
2. The server responds by sending a Hello message to the client. This message includes:
 - The algorithm to use. The server selected this from the list sent by the client.
 - A random number, which will be used to generate the keys.

3. The server sends its certificate to the client.
4. The client authenticates the server by checking the validity of the server's certificate, the issuer CA, and optionally, by checking that the host name of the server matches the subject DN. The client sends a Session ID for session caching.
5. The client generates a random value ("pre-master secret"), encrypts it using the server's public key, and sends it to the server.
6. The server uses its private key to decrypt the message to retrieve the pre-master secret.
7. The client and server separately calculate the keys that will be used in the SSL session.

These keys are not sent to each other because the keys are calculated based on the pre-master secret and the random numbers, which are known to each side. The keys include:

- Encryption key that the client uses to encrypt data before sending it to the server
- Encryption key that the server uses to encrypt data before sending it to the client
- Key that the client uses to create a message digest of the data
- Key that the server uses to create a message digest of the data

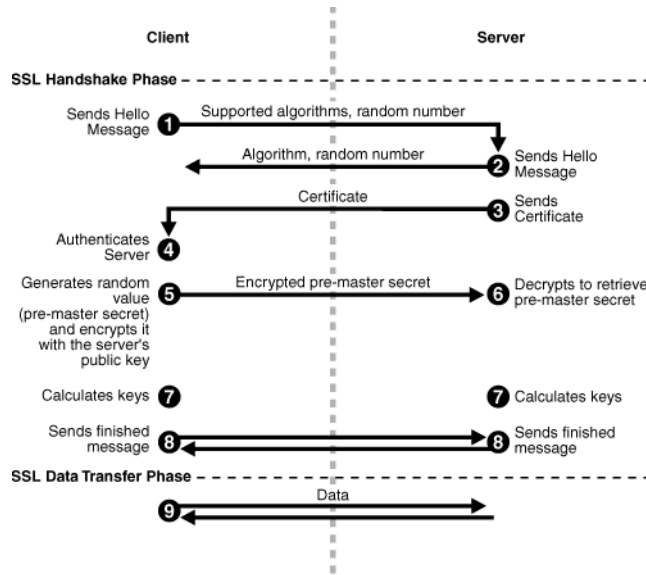
The encryption keys are symmetric, that is, the same key is used to encrypt and decrypt the data.

8. The client and server send a `Finished` message to each other. These are the first messages that are sent using the keys generated in the previous step (the first "secure" messages).

The `Finished` message includes all the previous handshake messages that each side sent. Each side verifies that the previous messages that it received match the messages included in the `Finished` message. This checks that the handshake messages were not tampered with.

9. The client and server now transfer data using the encryption and hashing keys and algorithms.

Figure 6-1 SSL Handshake



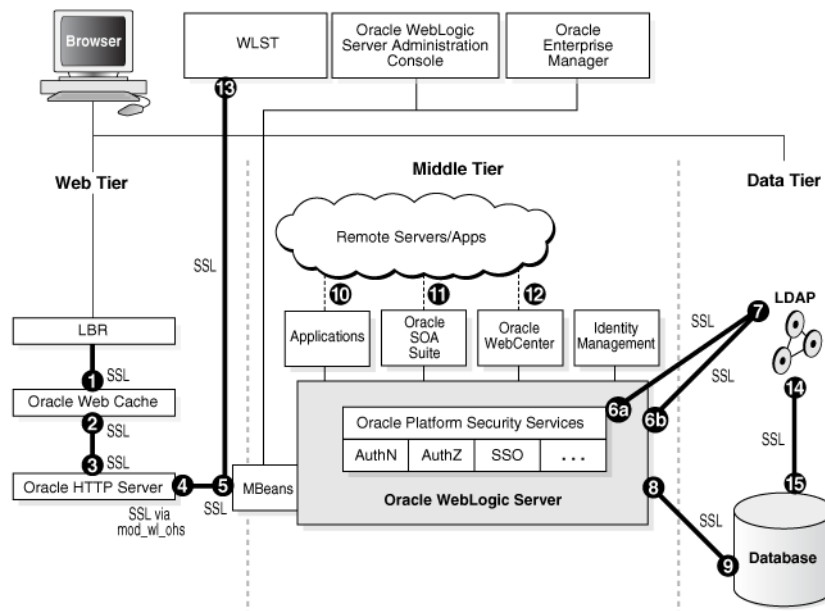
6.2 About SSL in Oracle Fusion Middleware

This section introduces SSL in Oracle Fusion Middleware. It contains these topics:

- [SSL in the Oracle Fusion Middleware Architecture](#)
- [Keystores and Oracle Wallets](#)
- [Authentication Modes](#)
- [Tools for SSL Configuration](#)

6.2.1 SSL in the Oracle Fusion Middleware Architecture

Figure 6-2 SSL in Oracle Fusion Middleware



Notes:

- In [Figure 6–2](#), the label "Oracle Enterprise Manager" refers to the Fusion Middleware Control user interface.
 - Other administrative tools are available for specific tasks.
-

In the Oracle Fusion Middleware architecture shown in [Figure 6–2](#), the numbered circles represent the endpoints that can be SSL-enabled. For configuration details about each endpoint, see:

1. [Section 6.4.2.1, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control"](#) and [Section 6.4.2.2, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using WLST"](#)
2. [Section 6.4.2.3, "Enable SSL for Outbound Requests from Oracle HTTP Server"](#)
3. [Section 6.5.1.1, "Inbound SSL to Oracle WebLogic Server"](#)
4. Outbound connections to the LDAP server can originate from Oracle Platform Security Services or from Oracle WebLogic Server:
 - a. [Section 6.5.1.2.1, "Outbound SSL from Oracle Platform Security Services to LDAP"](#)
 - b. [Section 6.5.1.2.3, "Outbound SSL from LDAP Authenticator to LDAP"](#)
5. [Section 6.6.1.2, "SSL-Enable a Data Source"](#)
6. [Section 6.6.1.1, "SSL-Enable Oracle Database"](#)
7. [Section 6.5.2, "Client-Side SSL for Applications"](#)
8. [Section 6.5.1.2, "Outbound SSL from Oracle WebLogic Server"](#)
9. [Section 6.6.1.1, "SSL-Enable Oracle Database"](#)

Keystores and Wallets

Keystores and wallets are central to SSL configuration and are used to store certificates and keys.

For details, see [Section 6.2.2, "Keystores and Oracle Wallets."](#)

6.2.2 Keystores and Oracle Wallets

Oracle Fusion Middleware 12c (12.1.2) supports different types of keystores for keys and certificates:

- JKS-based keystore and truststore
Oracle WebLogic Server uses JKS keystores in upgraded environments.
JDK's `keytool` utility manages JKS keystores and certificates.
- Oracle wallet
System components like Oracle HTTP Server use the Oracle wallet.
Use Fusion Middleware Control, or the command-line WLST and `orapki` interfaces, to manage wallets and their certificates for system components. You can use either the Fusion Middleware Control or WLST to SSL-enable the listeners for these components.
- OPSS Keystore Service (KSS) keystore and truststore

The Keystore Service provides an alternative mechanism to manage keys and certificates. Oracle WebLogic Server uses KSS keystores out-of-the-box in new 12c (12.1.2) installations.

Use Fusion Middleware Control or WLST to manage KSS keystores and their certificates. Use the WebLogic console to SSL-enable the listeners for WebLogic servers.

For more information about these types of stores, and when to use which type of store, see [Section 6.1.3, "Keystores and Wallets"](#).

See Also: [Section 7.1, "Key and Certificate Storage in Oracle Fusion Middleware"](#) for keystore management

6.2.3 Authentication Modes

The following authentication modes are supported:

- In *no-authentication mode*, neither server nor client are required to authenticate. Other names for this mode include Anonymous SSL/No Authentication/Diffie-Hellman. This mode is considered unsecured.
- In *server authentication mode*, a server authenticates itself to a client. This mode is also referred to as One-way SSL/Server Authentication.
- In *mutual authentication mode*, a client authenticates itself to a server and that server authenticates itself to the client. This mode is also known as Two-way SSL/Client Authentication.
- In *optional client authentication mode*, the server authenticates itself to the client, but the client may or may not authenticate itself to the server. Even if the client does not authenticate itself, the SSL session still goes through.

6.2.4 Tools for SSL Configuration

Oracle Fusion Middleware uses two kinds of configuration tools, common and advanced.

Common Tools

- Fusion Middleware Control
- WLST command-line interface
- Oracle WebLogic Server Administration Console
- `keytool` command-line tool

These tools allow you to configure SSL and manage Oracle Wallet/JKS keystore for any listener or component in Oracle Fusion Middleware.

The first three tools on this list are usable when the component is associated with the application server domain (when the component is not a stand-alone installation).

Advanced Tools

- `orapki` command-line interface

This tool is needed to manage wallets for certain stand-alone installations.

In addition, this tool allows you to configure advanced features like managing file-based CRLs, PKCS11-based wallets, and so on.

See Also: [Section 7.1, "Key and Certificate Storage in Oracle Fusion Middleware"](#) for keystore management

6.3 Configuring SSL for Configuration Tools

Several tools are available for Oracle Fusion Middleware configuration. This section describes how to configure SSL for these tools to enable them to connect to an SSL-enabled Oracle WebLogic Server.

See Also: [Section 6.5.1.1](#) for details about enabling inbound SSL on Oracle WebLogic Server.

For a list of all the configuration tools, see [Section 6.2.4, "Tools for SSL Configuration."](#)

This section contains these topics:

- [Oracle Enterprise Manager Fusion Middleware Control](#)
- [Oracle WebLogic Server Administration Console](#)
- [WLST Command-Line Tool](#)

6.3.1 Oracle Enterprise Manager Fusion Middleware Control

Take these steps:

- Ensure that the SSL port is enabled on the Oracle WebLogic Server instance on which Fusion Middleware Control is deployed, and that the browser (from which you will launch Fusion Middleware Control) trusts the server certificate.
- Now launch Fusion Middleware Control using an SSL-based URL, in the format `https://host:port`.

6.3.2 Oracle WebLogic Server Administration Console

Ensure that the SSL port is enabled on the Oracle WebLogic Server instance. Now launch the administration console by providing the SSL port in the URL. You may get a warning that the certificate is not trusted; accept this certificate and continue.

6.3.3 WLST Command-Line Tool

For details about configuring SSL for WLST, take these steps:

1. Launch the WLST shell.
2. Execute the WLST command:

```
help('connect')
```

Follow the instructions described in the help text to set up the WLST shell in SSL mode.

See Also: [Section 6.9](#) for details about using WLST.

6.4 Configuring SSL for the Web Tier

This section contains these topics:

- [Configuring Load Balancers](#)
- [Enabling SSL for Oracle HTTP Server Virtual Hosts](#)

Note:

- This discussion applies to the Web Tier in the context of an Oracle WebLogic Server domain.
 - The order in which these topics appear should not be confused with the sequence in which SSL is enabled (which varies depending on topology). Rather, they are arranged in order starting with the most front-ending component.
-
-

6.4.1 Configuring Load Balancers

Use the instructions specific to your load-balancing device to configure load balancers in your Oracle Fusion Middleware environment.

6.4.2 Enabling SSL for Oracle HTTP Server Virtual Hosts

This section shows how to manage SSL configuration for Oracle HTTP Server virtual hosts operating in an Oracle WebLogic Server environment.

Note: For Oracle HTTP Server in standalone mode, see *Administering Oracle HTTP Server*.

For inbound traffic:

- [Section 6.4.2.1](#) (using Fusion Middleware Control)
- [Section 6.4.2.2](#) (using WLST)

For outbound traffic:

- [Section 6.4.2.3](#) (using either Fusion Middleware Control or WLST)

6.4.2.1 Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control

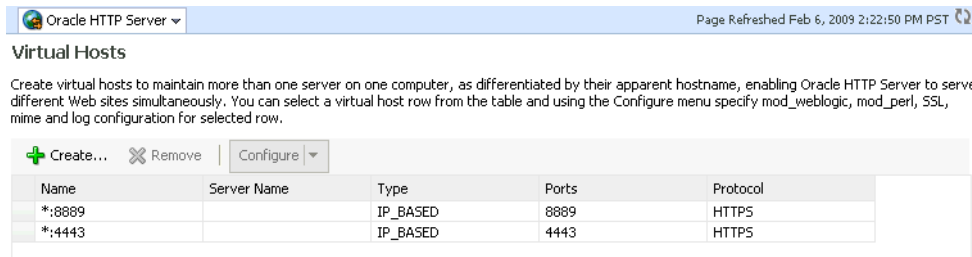
You can SSL-enable inbound traffic to Oracle HTTP Server virtual hosts using these steps:

1. Select the Oracle HTTP Server instance in the navigation pane on the left.
2. Create a wallet, if necessary, by navigating to **Oracle HTTP Server**, then **Security**, then **Wallets**.

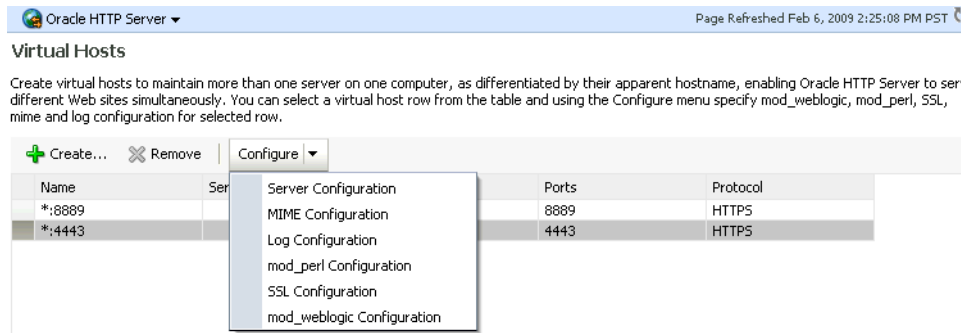
For details about wallet creation and maintenance, see [Chapter 7, "Managing Keystores, Wallets, and Certificates"](#).

3. Navigate to **Oracle HTTP Server**, then **Administration**, then **Virtual Hosts**.

This page shows what hosts are currently configured, and whether they are configured for http or https.



4. Select the virtual host you wish to update, and click **Configure**, then **SSL Configuration**. (Note: If creating a new virtual host, see *Administering Oracle HTTP Server*.)

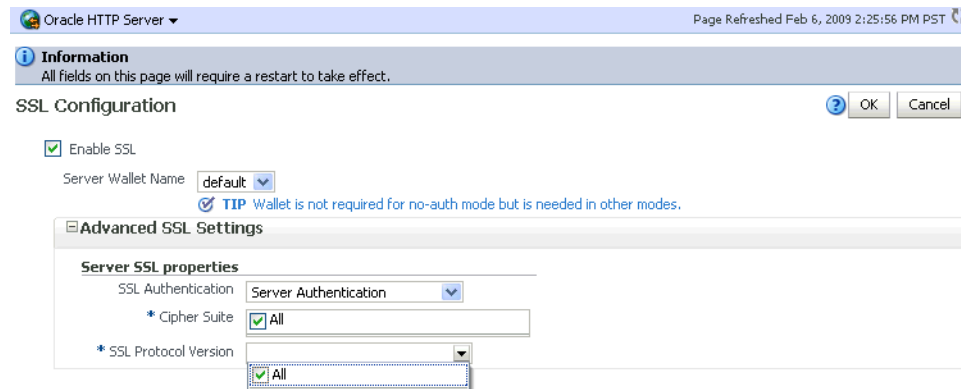


The SSL Configuration page appears.

5. You can convert an https port to http by simply unchecking **Enable SSL**.

To configure SSL for a virtual host that is currently using http:

- Check the **Enable SSL** box.
- Select a wallet from the drop-down list.



- From the Server SSL properties, select the SSL authentication type, cipher suites to use, and the SSL protocol version.

Note: The default values are appropriate in most situations.

Note: ■ This assumes that the certificate is available in Fusion Middleware Control. If it was created through orapki, import it first as explained in [Section 7.4.4.9](#).

- The choice of authentication type determines the available cipher suites, and the selected cipher suites determine the available protocol versions. For more information about ciphers and protocol versions, see [Section 6.9.16](#).
-
-

6. Click **OK** to apply the changes.
7. On Windows platforms only, open Windows Explorer and navigate to your cwallet.sso file. Under properties, security, add **SYSTEM** in "group or user names".
8. Restart the Oracle HTTP Server instance by navigating to **Oracle HTTP Server**, then **Control**, then **Restart**.
9. Open a browser session and connect to the port number that was SSL-enabled.

6.4.2.2 Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using WLST

Take these steps:

1. Determine the virtual hosts for this **Oracle HTTP Server** instance by running the following command:

```
listListeners('inst1','ohs1'#)
```

This command lists all the virtual hosts for this instance. Select the one that needs to be configured for SSL. For example, you can select vhost1. For details about this command, see *WLST Command Reference for WebLogic Server*.

2. Configure the virtual host with SSL properties:

```
configureSSL('inst1',
### 'ohs1',
### 'ohs',
### 'vhost1')
```

Note:

- `configureSSL` uses defaults for all SSL attributes; see [Table 6-4](#) for details.
 - You could also specify a properties file as a parameter to `configureSSL`. See [Table 6-3](#) for details about the parameters. See `configureSSL` in the *WLST Command Reference for WebLogic Server* for examples of how to use a properties file.
 - For details about this command, see [Section 6.9](#).
-
-

3. On Windows platforms only, open Windows Explorer and navigate to your cwallet.sso file. Under properties, security, add **SYSTEM** in "group or user names".

6.4.2.3 Enable SSL for Outbound Requests from Oracle HTTP Server

You enable SSL for outbound requests from Oracle HTTP Server by configuring `mod_wl_ohs`.

One-way SSL

The steps are as follows:

1. Generate a custom keystore for Oracle WebLogic Server (see [Section 6.5.1](#)) containing a certificate.
2. Import the certificate used by Oracle WebLogic Server from Step 1 into the Oracle HTTP Server wallet as a trusted certificate. You can use any available utility such as WLST or Fusion Middleware Control for this task. (*Note:* The wallet mentioned here is the one that the Oracle HTTP Server listen port uses for SSL. The trusted (root) CA certificate that signed the Oracle WebLogic Server certificate must exist in this wallet. For details on importing trusted certificates see [Section 7.4.7.5](#).)
3. Edit the Oracle HTTP Server configuration file `DOMAIN_HOME/config/fmwconfig/components/OHS/instance_name/ssl.conf` and add the following line to the SSL configuration under `mod_weblogic`:

```
WlSSLWallet##"${DOMAIN_HOME}/config/fmwconfig/components/COMPONENT_
TYPE/COMPONENT_NAME/kestores/default"
```

where `default` is the name of the Oracle HTTP Server wallet in Step 2.

Here is an example of how the configuration should look:

```
<IfModule#mod_weblogic.c>
WebLogicHost#myweblogic.server.com
WebLogicPort#7002
MatchExpression#*.jsp
SecureProxy#On
WlSSLWallet#"${DOMAIN_
HOME}/config/fmwconfig/components/OHS/ohs1/kestores/default"
</IfModule>
```

Save the file and exit.

4. On Windows platforms only, open Windows Explorer and navigate to your `cwallet.sso` file. Under properties, security, add `SYSTEM` in "group or user names".
5. Restart Oracle HTTP Server to activate the changes. See *Administering Oracle HTTP Server* for details.
6. Ensure that your Oracle WebLogic Server instance is configured to use the custom keystore generated in Step 1, and that the alias points to the alias value used in generating the certificate. Restart the Oracle WebLogic Server instance.

Two-way SSL

`mod_wl_ohs` also supports two-way SSL communication. To configure two-way SSL:

1. Perform Steps 1 through 4 of the preceding procedure for one-way SSL.
2. Generate a trust store, `trust.jks`, for Oracle WebLogic Server.

The keystore created for one-way SSL (Step 1 of the preceding procedure) could also be used to store trusted certificates, but it is recommended that you create a separate truststore for this procedure.
3. Export the user certificate from the Oracle HTTP Server wallet, and import it into the truststore created in Step 2.

You can use any available utility such as WLST or Fusion Middleware Control for export, and the `keytool` utility for import.

4. From the Oracle WebLogic Server Administration Console, select the **Keystores** tab for the server being configured.
5. Set the custom trust store with the `trust.jks` file location of the trust store that was created in Step 2 (use the full name).
6. Set the keystore type as JKS, and set the passphrase used to create the keystore.
7. Under the **SSL** tab, ensure that Trusted Certificate Authorities is set as **from Custom Trust Keystore**.
8. Ensure that Oracle WebLogic Server is configured for two-way SSL. For details, see "Configuring SSL" in *Administering Security for Oracle WebLogic Server*.

6.5 Configuring SSL for the Middle Tier

Using SSL in the middle tier includes:

- SSL-enabling the application server
- SSL-enabling components and applications running on the application server

This section addresses mid-tier SSL configuration and contains these topics:

- [Configuring SSL for Oracle WebLogic Server](#)
- [Client-Side SSL for Applications](#)

6.5.1 Configuring SSL for Oracle WebLogic Server

This section describes configuration for the application server.

6.5.1.1 Inbound SSL to Oracle WebLogic Server

For information and details about implementing SSL to secure Oracle WebLogic Server, see the following topics in *Administering Security for Oracle WebLogic Server*:

- "Configuring Oracle OPSS Keystore Service "
- "Configuring SSL"

6.5.1.2 Outbound SSL from Oracle WebLogic Server

This section describes how to SSL-enable outbound connections from Oracle WebLogic Server.

- [Outbound SSL from Oracle Platform Security Services to LDAP](#)
- [Outbound SSL from Oracle Platform Security Services to Oracle Database](#)
- [Outbound SSL from LDAP Authenticator to LDAP](#)
- [Outbound SSL to Database](#)

6.5.1.2.1 Outbound SSL from Oracle Platform Security Services to LDAP This section explains how to configure SSL (needs server- and client-side) for policy store and credential store connections to an LDAP directory. It supports anonymous and one-way SSL.

See *Securing Applications with Oracle Platform Security Services* for details about the `jps-config.xml` file referenced in this section.

Anonymous SSL (Server-side)

Start the LDAP server in anonymous authentication mode.

Consult your LDAP server documentation for information on this task.

Anonymous SSL (Client-side)

In your `jps-config.xml` file, you must set the protocol to `ldaps` and specify the appropriate port for the property `ldap.url`. This information needs to be updated for policy store, credential store, key store and any other service instances that use `ldap.url`.

One-Way SSL (Client-side)

The following must be in place for the client-side configuration:

1. The JVM needs to know where to find the trust store that it uses to trust certificates from LDAP. You do this by setting:

```
-Djavax.net.ssl.trustStore=path_to_jks_file
```

This property is added either to the JavaSE program, or to the server start-up properties in a JavaEE environment.

2. In your `jps-config.xml` file, you must set the protocol to `ldaps` and specify the appropriate port for the property `ldap.url`. This information needs to be updated for policy store, credential store, key store and any other service instances that use `ldap.url`.
3. Using **keytool**, import the LDAP server's certificate into the trust store specified in step 1.

6.5.1.2.2 Outbound SSL from Oracle Platform Security Services to Oracle Database You can set up a one-way or two-way SSL connection to a database-based OPSS security store.

For details about configuring the database server and clients, see *Securing Applications with Oracle Platform Security Services*.

6.5.1.2.3 Outbound SSL from LDAP Authenticator to LDAP For details about outbound SSL to LDAP directories, see "How SSL Certificate Validation Works in WebLogic Server" in *Administering Security for Oracle WebLogic Server*.

6.5.1.2.4 Outbound SSL to Database See "Configuring Secure Sockets Layer Authentication" in the *Oracle Advanced Security Administrator's Guide* at http://docs.oracle.com/cd/E11882_01/network.112/e10746/asossl.htm for more information about configuring SSL for the database listener.

6.5.2 Client-Side SSL for Applications

For information on how to write SSL-enabled applications, see "Using SSL Authentication in Java Clients" in *Developing Applications with the WebLogic Security Service*.

For best practices, refer to [Section 6.8.2, "Best Practices for Application Developers."](#)

6.6 Configuring SSL for the Data Tier

This section contains this topic:

- [Configuring SSL for the Database](#)

6.6.1 Configuring SSL for the Database

This section contains these topics:

- [SSL-Enable Oracle Database](#)
- [SSL-Enable a Data Source](#)

6.6.1.1 SSL-Enable Oracle Database

Take these steps to SSL-enable Oracle database:

1. Create a root CA and a certificate for the DB. Here is an example:

Note: Self-signed certificates are not recommended for production use. For information about obtain production wallets, see [Section 7.4.8.3, "Changing a Self-Signed Wallet to a Third-Party Wallet."](#)

```
mkdir#root
mkdir#server
#
##Create#root#wallet,#add#self-signed#certificate#and#export
orapki#wallet#create#-wallet#./root#-pwd#password
orapki#wallet#add#-wallet#./root#-dn#CN=root_test,C=US#-keysize#2048#-self_
signed#-validity#3650#-pwd#password
orapki#wallet#display#-wallet#./root#-pwd#password
orapki#wallet#export#-wallet#./root#-dn#CN=root_test,C=US#-cert#
./root/b64certificate.txt#-pwd#password
#
#Create#server#wallet,#add#self-signed#certificate#and#export
orapki#wallet#create#-wallet#./server#-pwd#password
orapki#wallet#add#-wallet#./server#-dn#CN=server_test,C=US#-keysize#2048#-pwd#
password
orapki#wallet#display#-wallet#./server#-pwd#password
orapki#wallet#export#-wallet#./server#-dn#CN=server_test,C=US#-request#
./server/creq.txt#-pwd#password
#
##Import#trusted#certificates
orapki#cert#create#-wallet#./root#-request#./server/creq.txt#-cert#
./server/cert.txt#-validity#3650#-pwd#password
orapki#cert#display#-cert#./server/cert.txt#-complete
orapki#wallet#add#-wallet#./server#-trusted_cert#-cert#
./root/b64certificate.txt#-pwd#password
orapki#wallet#add#-wallet#./server#-user_cert#-cert#./server/cert.txt#-pwd#
password
orapki#wallet#create#-wallet#./server#-auto_login#-pwd#password}}
```

2. Update listener.ora, sqlnet.ora, and tnsnames.ora for the database.

- a. This example shows the default listener.ora:

```
SID_LIST_LISTENER#=
(SID_LIST#=(SID_DESC#=(SID_NAME#=#PLSExtProc)(ORACLE_HOME#=#/path_to_O_
H)(PROGRAM#=#extproc)))
LISTENER#=(DESCRIPTION_LIST#=(DESCRIPTION#=
(ADDRESS#=#(PROTOCOL#=#IPC)(KEY#=#EXTPROC1))
(ADDRESS#=#(PROTOCOL#=#TCP)(HOST#=#mynode.mycorp.com)(PORT#=#1521))
(ADDRESS#=#(PROTOCOL#=#TCPS)(HOST#=#mynode.mycorp.com)(PORT#=#2490))
))
#
```

```

WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/wallet_
location)))
#
SSL_CLIENT_AUTHENTICATION=FALSE}}

```

And here is an updated listener.ora file, illustrating a scenario with no client authentication:

```

SID_LIST_LISTENER#=
##(SID_LIST#=
####(SID_DESC#=
#####(GLOBAL_DBNAME#=#dbname)
#####(ORACLE_HOME#=#/path_to_O_H)
#####(SID_NAME#=#sid)
####)
##)
#
SSL_CLIENT_AUTHENTICATION = FALSE
#
WALLET_LOCATION =
##(SOURCE =
####(METHOD = FILE)
####(METHOD_DATA =
#####(DIRECTORY = /wallet_path)
####)
##)
#
LISTENER#=
##(DESCRIPTION_LIST#=
####(DESCRIPTION#=
#####(ADDRESS#=#(PROTOCOL#=#IPC)(KEY#=#EXTPROC1521))
####)
####(DESCRIPTION#=
#####(ADDRESS#=#(PROTOCOL#=#TCP)(HOST#=#mynode.mycorp.com)(PORT#=#1521))
####)
####(DESCRIPTION =
#####(ADDRESS = (PROTOCOL = TCPS)(HOST = mycorp.com)(PORT = 2490))
####)
##)

```

Note that the SSL port has been added.

- b. Likewise, a modified sqlnet.ora file may look like this:

```

NAMES.DIRECTORY_PATH=(TNSNAMES,#EZCONNECT)
SQLNET.AUTHENTICATION_SERVICES=(BEQ,TCPS,NTS)
WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/directory)))
SSL_CLIENT_AUTHENTICATION=FALSE

```

- c. A modified tnsnames.ora file may look like this:

```

OID#=
##(DESCRIPTION#=
####(ADDRESS#=#(PROTOCOL#=#TCP)(HOST#=#mynode.mycorp.com)(PORT#=#1521))
####(CONNECT_DATA#=
#####(SERVER#=#DEDICATED)
#####(SERVICE_NAME#=#mynode.mycorp.com)
####)
##)
###
SSL =
##(DESCRIPTION =

```

```
####(ADDRESS_LIST =
#####(ADDRESS = (PROTOCOL = TCPS)(HOST = mynode.mycorp.com)(PORT = 2490))
####)
####(CONNECT_DATA =
#####(SERVICE_NAME =#mynode.mycorp.com)
#####or
#####(SID#=#mynode.mycorp.com)
####)
####(SECURITY=(SSL_SERVER_CERT_DN=\"CN=server_test,C=US\"))
##)
```

3. Test the connection to the database using the new connect string. For example:

```
$#tnsping#ssl
$#sqlplus#username/password@ssl
```

See Also: The chapter "Configuring Secure Sockets Layer Authentication" in the *Oracle Database Advanced Security Administrator's Guide*.

6.6.1.2 SSL-Enable a Data Source

Take these steps to configure your data sources on Oracle WebLogic Server to use SSL.

1. Create a truststore and add the root certificate (which is created when SSL-enabling the database) as a trusted certificate to the truststore.
2. In the Oracle WebLogic Server Administration Console, navigate to the **Connection pool** tab of the data source that you are using.

Note: The data source can be an existing source such as an Oracle WebCenter Portal data source, or a new data source. See *Creating a JDBC Data Source in Administering JDBC Data Sources for Oracle WebLogic Server* for details.

The properties you need to specify in the **JDBC Properties** text box depend on the type of authentication you wish to configure.

- If you will require client authentication (two-way authentication):

```
javax.net.ssl.keyStore=..(password#of#the#keystore)
#####javax.net.ssl.keyStoreType=JKS
#####javax.net.ssl.keyStorePassword=..(password#of#the#keystore)
#####javax.net.ssl.trustStore=..(the#truststore#location#on#the#disk)
#####javax.net.ssl.trustStoreType=JKS
#####javax.net.ssl.trustStorePassword=..(password#of#the#truststore)
```

- If you will require no client authentication:

```
javax.net.ssl.trustStore=..(the#truststore#location#on#the#disk)
#####javax.net.ssl.trustStoreType=JKS
#####javax.net.ssl.trustStorePassword=..(password#of#the#truststore)
```

3. In the URL text box, enter the JDBC connect string. Ensure that the protocol is TCPS and that SSL_SERVER_CERT_DN contains the full DN of the database certificate.

Use the following syntax if tnsnames.ora uses "SERVICE_NAME":

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=TCPS)(HOST=host-name)(PORT=port-number)))(CONNECT_
```

```
DATA=(SERVICE_NAME=service))(SECURITY=(SSL_SERVER_CERT_DN="CN=server_
test,C=US")))
```

Use the following syntax if tnsnames.ora uses "SID":

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=TCPS)(HOST=host-name)(PORT=port-number)))(CONNECT_
DATA=(SID=service))(SECURITY=(SSL_SERVER_CERT_DN="CN=server_test,C=US")))
```

4. Test and verify the connection. Your data source is now configured to use SSL.

6.7 Advanced SSL Scenarios

This section explains how to handle additional SSL configuration scenarios beyond the basic topologies described earlier:

- [Hardware Security Modules and Accelerators](#)
- [CRL Integration with SSL](#)
- [Oracle Fusion Middleware FIPS 140-2 Settings](#)

For details and examples of the commands used in this section see [Section 6.9](#).

6.7.1 Hardware Security Modules and Accelerators

A Hardware Security Module (HSM) is a physical plug-in card or an external security device that can be attached to a computer to provide secure storage and use of sensitive content.

Note: This discussion applies only to Oracle HTTP Server, which is a system component supporting HSM.

Oracle Fusion Middleware supports PKCS#11-compliant HSM devices that provide a secure storage for private keys.

Take these steps to implement SSL for a component using a PKCS#11 wallet:

1. Install the HSM libraries on the machine where the component is running. This is a one-time task and is device-dependent.
2. Next, create a wallet using the `orapki` command-line tool. Note the following:
 - a. Choose PKCS11 as the wallet type.
 - b. Specify the device-specific PKCS#11 library used to communicate with the device. This library is part of the HSM software.

On Linux, the library is located at:

```
For#LunaSA#(Safenet):##/usr/lunasa/lib/libCryptoki2.so
For#nCipher:##/opt/nfast/toolkits/pkcs11/libcknfast.so
```

On Windows, the library is located at:

```
For#LunaSA#(Safenet):#C:\Program#Files\LunaSA\cryptoki.dll
```

3. Now follow the standard procedure for obtaining third-party certificates: create a certificate request; get the request approved by a Certificate Authority (CA); and install the certificate signed by that CA.

The wallet you set up is used like any other wallet.

4. Verify the wallet with the `orapki` utility. Use the following command syntax:

```
orapki#wallet#p11_verify#[-wallet#[wallet]]#[-pwd#password]
```

See Also: [Appendix G, "orapki"](#) for details about `orapki`

5. Configure SSL on your component listener using the `configureSSL WLST` command, providing a properties file as input. Your properties file should specify the full path of the PKCS#11 wallet directory on the machine where the component is running. (*Note:* Do not save the PKCS#11 wallet in the instance home directory. Only wallets created and managed through Fusion Middleware Control or WLST should reside in the instance home.)

A sample properties file could look like this:

```
SSLEnabled=true
AuthenticationType=Server
PKCS11Wallet=/tmp/lunasa/wallet#
```

Note: You must use the WLST command `configureSSL` to configure the PKCS11 wallet. You cannot do this task using Fusion Middleware Control or any other tool.

6.7.2 CRL Integration with SSL

Note:

- This discussion applies only to Oracle HTTP Server in the context of an Oracle WebLogic Server environment. For SSL configuration in standalone components, see *Administering Oracle HTTP Server*.
 - Manage CRL validation through WLST. You cannot perform this task through Fusion Middleware Control.
-

Components that use SSL can optionally turn on certificate validation using a certificate revocation list (CRL). This allows them to validate the peer certificate in the SSL handshake and ensure that it is not on the list of revoked certificates issued by the Certificate Authority (CA).

This section describes how to configure a component to use CRL-based validation, and how to create and set up CRLs on the file system.

6.7.2.1 Configuring CRL Validation for a Component

Configure SSL on your component listener using the `configureSSL WLST` command, providing a properties file as input.

The properties file must be set up as follows:

1. The `CertValidation` attribute must be set to `url`.
2. The `CertValidationPath` attribute must be of the form `file://file_path` or `dir://directory_path`.
 - Use the first format if you are using a single CRL file for certificate validation. This CRL file should contain a concatenation of all CRLs.
 - Use the second format if you are specifying a directory path that contains multiple CRL files in hashed form.

See [Section 6.7.2.2, "Manage CRLs on the File System"](#) on how to create CRLs in hashed form.

In this example, the properties file specifies a single CRL file:

```
SSLEnabled=true
AuthenticationType=Server
CertValidation=crl
KeyStore=ohs1#
CertValidationPath=file:///tmp/file.crl
```

In this example, the properties file specifies a directory path to multiple CRL files:

```
SSLEnabled=true
AuthenticationType=Server
KeyStore=ohs1#
CertValidation=crl
CertValidationPath=dir:///tmp
```

6.7.2.2 Manage CRLs on the File System

Note: LDAP-based CRLs or CRL distribution points are not supported.

You use the `orapki` command-line tool to manage CRLs on the file system. For details on this topic, see [Section G.1.5, "Managing Certificate Revocation Lists \(CRLs\) with orapki Utility."](#)

CRL Renaming to Hashed Form

If specifying a CRL storage location, the CRL must be renamed. This enables CRLs to be loaded in an efficient manner at runtime. This operation creates a symbolic link to the actual CRL file. On Windows, it copies the CRL to a file with a new name.

To rename a CRL:

```
orapki#crl#hash#
[-crl#[url|filename]]#[-wallet#wallet]#[-symlink#directory]#
[-copy#directory]#[-summary]#[-pwd#password]
```

For example:

```
orapki#crl#hash#-crl#nzcrl.txt#-symlink#wltdir#-pwd#password
```

If the CRL file name is specified at runtime, multiple CRLs can be concatenated in that file. The CRL created in this example is in Base64 format, and you can use a text editor to concatenate the CRLs.

CRL Creation

Note: CRL creation and Certificate Revocation are for test purposes and only used in conjunction with self-signed certificates. For production use, obtain production certificates from well-known CAs and obtain the CRLs from those authorities.

To create a CRL:

```
orapki#crl#create#
```

```
[-crl#[url|filename]]#[-wallet#[cawallet]]#[-nextupdate#[days]]#[-pwd#password]
```

For example:

```
orapki#crl#create#
-crl#nzcrl.crl#-wallet#rootwlt#-nextupdate#3650#-pwd#password
```

Certificate Revocation

Revoking a certificate adds the certificate's serial number to the CRL.

To revoke a certificate:

```
orapki#crl#revoke#
[-crl#[url|filename]]#[-wallet#[cawallet]]#[-cert###[revokecert]]#[-pwd#password]
```

For example:

```
orapki#crl#revoke#
-crl#nzcrl.txt#-wallet#rootwlt#-cert#cert.txt#-pwd#password
```

6.7.2.3 Test a Component Configured for CRL Validation

To test that a component is correctly configured for CRL validation, take these steps:

1. Set up a wallet with a certificate to be used in your component.
2. Generate a CRL with this certificate in the revoked certificates list. Follow the steps outlined in [Section 6.7.2.2, "Manage CRLs on the File System."](#)
3. Configure your component to use this CRL. Follow the steps outlined in [Section 6.7.2.1, "Configuring CRL Validation for a Component."](#)
4. The SSL handshake should fail when this revoked certificate is used.

6.7.3 Oracle Fusion Middleware FIPS 140-2 Settings

This section describes how to configure Oracle Fusion Middleware components to comply with the FIPS 140-2 advanced security standard. Topics include:

- [FIPS-Configurable Products](#)
- [Setting the SSLFIPS_140 Parameter](#)
- [Selecting Cipher Suites](#)
- [Other Configuration Parameters](#)

See Also: For more information about this standard, refer to the Cryptographic Modules Validation Program Web site at:

<http://csrc.nist.gov/groups/STM/index.html>

6.7.3.1 FIPS-Configurable Products

Any product using the Oracle SSL SDK can be configured to run in the FIPS mode. Specifically, you can configure Oracle HTTP Server in FIPS mode.

6.7.3.2 Setting the SSLFIPS_140 Parameter

You can configure these components to run in the FIPS mode by setting the SSLFIPS_140 parameter to TRUE in the `fips.ora` file:

```
SSLFIPS_140=TRUE
```

This file does not exist out-of-the-box and has to be created. Locate `fips.ora` either in the `$ORACLE_HOME/ldap/admin` directory, or in the directory pointed to by the `FIPS_HOME` environment variable.

The `SSLFIPS_140` parameter is set to `FALSE` by default. You must set it to `TRUE` for FIPS mode operation.

6.7.3.3 Selecting Cipher Suites

A cipher suite is a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, for example, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

Only the following cipher suites are approved for use in FIPS mode:

```
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
```

Any other ciphers should not be used while running in FIPS mode.

You can configure one or more of these ciphers using comma-separated values. These should be specified in the SSL properties file for the key 'Ciphers' in the `WLST#configureSSL` command, or through Fusion Middleware Control.

See [Section 6.9.16, "Properties Files for SSL"](#) for details about specifying the SSL properties file with the `configureSSL` command.

6.7.3.4 Other Configuration Parameters

The minimum key size for enabling FIPS mode is 1024 bits. You need to ensure that the keys used in FIPS mode are at least 1024 bits. This is because the certificate in the wallet used by components like Oracle HTTP Server must have a minimum public key size of 1024 bits.

You can only use wallets created using Oracle tools such as `orapki`. Third-party PKCS#12 wallet files cannot be used in FIPS mode.

6.8 Best Practices for SSL

This section outlines some best practices for Oracle Fusion Middleware component administrators and application developers. It contains these topics:

- [Best Practices for Administrators](#)
- [Best Practices for Application Developers](#)

6.8.1 Best Practices for Administrators

Best practices for system administrators include the following:

- Use self-signed wallets only in test environment. You should obtain a CA signed certificate in the wallet before moving to production environment. For details, see [Chapter 7, "Managing Keystores, Wallets, and Certificates."](#)
- It is recommended that components (at least in the Web tier) use certificates that have the system host name or virtual host or site name as the DN. This allows browsers to connect in SSL mode without giving unsettling warning messages.

- A minimum key size of 1024 bits is recommended for certificates used for SSL. Higher key size provides more security but at the cost of reduced performance. Pick an appropriate key size value depending on your security and performance requirements.
- Lack of trust is one of the most common reasons for SSL handshake failures. Ensure that the client trusts the server (by importing the server CA certificate into the client keystore) before starting SSL handshake. If client authentication is also required, then the reverse should also be true.

6.8.2 Best Practices for Application Developers

The following practices are recommended:

- Use Java Key Store (JKS) to store certificates for your Java EE applications.
- Externalize SSL configuration parameters like keystore path, truststore path, and authentication type in a configuration file, rather than embedding these values in the application code. This allows you the flexibility to change SSL configuration without having to change the application itself.

6.9 WLST Reference for SSL

WLST commands are available to manage Oracle wallets and to configure SSL for Oracle Fusion Middleware components.

Use the commands listed in [Table 6–1](#) and [Table 6–2](#) for this task.

See Also: [Section 7.2, "Command-Line Interface for Keystores and Wallets"](#) for important instructions on how to launch the WLST shell to run SSL-related commands. Do not launch the WLST interface from any other location.

Note: All WLST commands for SSL configuration must be run in online mode.

You can obtain help for each command by issuing:

```
help('command_name')
```

Certain commands require parameters like instance name, ias-component and process type. You can obtain this information with the command:

```
state('serverName')#[in#WebLogic#domain]
```

```
nmServerStatus(serverName='name', #serverType='type')#[in#Standalone#domain]
```

Table 6–1 WLST Commands for SSL Configuration

Use this command...	To...	Use with WLST...
configureSSL	Set the SSL attributes for a component listener.	Online
getSSL	Display the SSL attributes for a component listener.	Online

Table 6–2 WLST Commands for Oracle Wallet Management

Use this command...	To...	Use with WLST...
<code>addCertificateRequest</code>	Generate a certificate signing request in an Oracle wallet.	Online
<code>addSelfSignedCertificate</code>	Add a self-signed certificate to an Oracle wallet.	Online
<code>changeWalletPassword</code>	Change the password to an Oracle wallet.	Online
<code>createWallet</code>	Create an Oracle wallet.	Online
<code>deleteWallet</code>	Delete an Oracle wallet.	Online
<code>exportWallet</code>	Export an Oracle wallet to a file.	Online
<code>exportWalletObject</code>	Export an object (for example, a certificate) from an Oracle wallet to a file.	Online
<code>getWalletObject</code>	Display a certificate or other object present in an Oracle wallet.	Online
<code>importWallet</code>	Import an Oracle wallet from a file.	Online
<code>importWalletObject</code>	Import a certificate or other object from a file to an Oracle wallet.	Online
<code>listWalletObjects</code>	List all objects (such as certificates) present in an Oracle wallet.	Online
<code>listWallets</code>	List all Oracle wallets configured for a component instance.	Online
<code>removeWalletObject</code>	Remove a certificate or other object from a component instance's Oracle wallet.	Online

Note: WLST allows you to import certificates only in PEM format.

6.9.1 addCertificateRequest

Online command that generates a certificate signing request in an Oracle wallet.

6.9.1.1 Description

This command generates a certificate signing request in Base64 encoded PKCS#10 format in an Oracle wallet for a component instance (Oracle HTTP Server). To get a certificate signed by a certificate authority (CA), send the certificate signing request to your CA.

6.9.1.2 Syntax

```
addCertificateRequest('instName', 'compName', 'compType', 'walletName',
'password', 'DN', 'keySize')
```

Argument	Definition
<code>instName</code>	Specifies the name of the application server instance.
<code>compName</code>	Specifies the name of the component instance.
<code>compType</code>	Specifies the type of component. Valid value is 'ohs'.
<code>walletName</code>	Specifies the name of the wallet file.

Argument	Definition
password	Specifies the password of the wallet.
DN	Specifies the Distinguished Name of the key pair entry.
keySize	Specifies the key size in bits.

6.9.1.3 Example

The following command generates a certificate signing request with DN `cn=www.acme.com` and key size 1024 in wallet1, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#addCertificateRequest('inst1',#'ohs1',#
'ohs','wallet1',#'password',#'cn=www.acme.com',#'1024',)
```

6.9.2 addSelfSignedCertificate

Online command that adds a self-signed certificate.

6.9.2.1 Description

This command creates a key pair and wraps it in a self-signed certificate in an Oracle wallet for the specified component instance (Oracle HTTP Server). Only keys based on the RSA algorithm are generated.

6.9.2.2 Syntax

```
addSelfSignedCertificate('instName', 'compName', 'compType', 'walletName',
'password', 'DN', 'keySize')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file.
password	Specifies the password of the wallet.
DN	Specifies the Distinguished Name of the key pair entry.
keySize	Specifies the key size in bits.

6.9.2.3 Example

The following command adds a self-signed certificate with DN `cn=www.acme.com`, key size 1024 to wallet1, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#addSelfSignedCertificate('inst1',#'ohs1',#
'ohs','wallet1',#'password',#'cn=www.acme.com',#'1024')
```

6.9.3 changeWalletPassword

Online command that changes the password of an Oracle wallet.

6.9.3.1 Description

This command changes the password of an Oracle wallet for the specified component instance (Oracle HTTP Server). This command is only applicable to password-protected wallets.

6.9.3.2 Syntax

```
changeWalletPassword('instName', 'compName', 'compType',
'walletName', 'currPassword', 'newPassword')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the filename of the wallet.
currPassword	Specifies the current wallet password.
newPassword	Specifies the new wallet password.

6.9.3.3 Example

The following command changes the password for wallet1 from currpassword to newpassword for Oracle HTTP Server instance ohs1 in application server instance inst1:

```
wls:/mydomain/serverConfig>#changeWalletPassword('inst1',#'ohs1',#'ohs', 'wallet1',#
'currpassword',#'newpassword')
```

6.9.4 configureSSL

Online command that sets SSL attributes.

6.9.4.1 Description

This command sets the SSL attributes for a component listener. The attributes are specified in a properties file format (name=value). If a properties file is not provided, or it does not contain any SSL attributes, then default attribute values are used.

For details about the format of properties files, see [Section 6.9.16, "Properties Files for SSL."](#)

6.9.4.2 Syntax

```
configureSSL('instName', 'compName', 'compType', 'listener', 'filePath')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
listener	Specifies the name of the component listener to be configured for SSL.
filePath	Specifies the absolute path of the properties file containing the SSL attributes to set.

6.9.4.3 Examples

The following command configures SSL attributes specified in the properties file `/tmp/ssl.properties` for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`, for listener `listener1`:

```
wls:/mydomain/serverConfig>#configureSSL('inst1',#'ovd1',#'ovd',#
'listener1','/tmp/ssl.properties')
```

The following command configures SSL attributes without specifying a properties file. Since no file is provided, the default SSL attribute values are used:

```
wls:/mydomain/serverConfig>#configureSSL('inst1',#'ovd1',#'ovd',#'listener2')
```

6.9.5 createWallet

Online command that creates an Oracle wallet.

6.9.5.1 Description

This command creates an Oracle wallet for the specified component instance (Oracle HTTP Server). Wallets can be of password-protected or auto-login type. For wallet details, see [Chapter 7, "Managing Keystores, Wallets, and Certificates."](#)

6.9.5.2 Syntax

```
createWallet('instName', 'compName', 'compType', 'walletName', 'password')
```

Argument	Definition
<code>instName</code>	Specifies the name of the application server instance.
<code>compName</code>	Specifies the name of the component instance.
<code>compType</code>	Specifies the type of component. Valid value is 'ohs'.
<code>walletName</code>	Specifies the name of the wallet file to be created.
<code>password</code>	Specifies the wallet password.

6.9.5.3 Examples

The following command creates a wallet named `wallet1` with password `password`, for Oracle HTTP Server instance `ohs1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#createWallet('inst1',#'ohs1',#'ohs','wallet1',#
'password')
```

The following command creates an auto-login wallet named `wallet2` for Oracle WebCache instance `wc1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#createWallet('inst1',#'wc1',#'webcache','wallet2',#''')
```

6.9.6 deleteWallet

Online command that deletes an Oracle wallet.

6.9.6.1 Description

This command deletes an Oracle wallet for the specified component instance.

6.9.6.2 Syntax

```
deleteWallet('instName', 'compName', 'compType', 'walletName')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file to be deleted.

6.9.6.3 Example

The following command deletes a wallet named `wallet1` for Oracle HTTP Server instance `ohs1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#deleteWallet('inst1','#ohs1','#ohs','wallet1')
```

6.9.7 exportWallet

Online command that exports an Oracle wallet.

6.9.7.1 Description

This command exports an Oracle wallet, configured for a specified component instance, to files under the given directory. If the exported file is an auto-login only wallet, the file name is `cwallet.sso`. If it is password-protected wallet, two files are created—`ewallet.p12` and `cwallet.sso`.

6.9.7.2 Syntax

```
exportWallet('instName', 'compName', 'compType', 'walletName', 'password', 'path')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file.
password	Specifies the password of the wallet.
path	Specifies the absolute path of the directory under which the object is exported.

6.9.7.3 Examples

The following command exports auto-login wallet `wallet1` for Oracle HTTP Server instance `ohs1` to file `cwallet.sso` under `/tmp`:

```
wls:/mydomain/serverConfig>#exportWallet('inst1','#ohs1','#ohs',#  
'wallet1','','/tmp')
```

The following command exports password-protected wallet `wallet2` for Oracle HTTP Server instance `ohs1` to two files, `ewallet.p12` and `cwallet.sso`, under `/tmp`:

```
wls:/mydomain/serverConfig>#exportWallet('inst1','#ohs1','#ohs',#'wallet2',#  
'password','#/tmp')
```

6.9.8 exportWalletObject

Online command that exports a certificate or other wallet object to a file.

6.9.8.1 Description

This command exports a certificate signing request, certificate, certificate chain or trusted certificate present in an Oracle wallet to a file for the specified component instance. DN indicates the object to be exported.

6.9.8.2 Syntax

```
exportWalletObject('instName',#'compName',#'compType',#'walletName',#'password',#'type',#'path',#'DN')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file.
password	Specifies the password of the wallet.
type	Specifies the type of wallet object to be exported. Valid values are 'CertificateRequest', 'Certificate', 'TrustedCertificate' or 'TrustedChain'.
path	Specifies the absolute path of the directory under which the object is exported as a file base64.txt.
DN	Specifies the Distinguished Name of the wallet object being exported.

6.9.8.3 Examples

The following command exports a certificate signing request with DN `cn=www.acme.com` in `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`. The certificate signing request is exported under the directory `/tmp`:

```
wls:/mydomain/serverConfig>#exportWalletObject('inst1',#'ohs1',#'ohs','wallet1',#'password',#'CertificateRequest',#'/tmp','cn=www.acme.com')
```

The following command exports a certificate with DN `cn=www.acme.com` in `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`. The certificate or certificate chain is exported under the directory `/tmp`:

```
wls:/mydomain/serverConfig>#exportWalletObject('inst1',#'ohs1',#'ohs','wallet1',#'password',#'Certificate',#'/tmp','cn=www.acme.com')
```

The following command exports a trusted certificate with DN `cn=www.acme.com` in `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`. The trusted certificate is exported under the directory `/tmp`:

```
wls:/mydomain/serverConfig>#exportWalletObject('inst1',#'ohs1',#'ohs','wallet1',#'password',#'TrustedCertificate',#'/tmp','cn=www.acme.com')
```

The following command exports a certificate chain with DN `cn=www.acme.com` in `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`. The certificate or certificate chain is exported under the directory `/tmp`:

```
wls:/mydomain/serverConfig>#exportWalletObject('inst1',#'ohs1',#'ohs','wallet1',#'password',#'TrustedChain',#'/tmp','cn=www.acme.com')
```

6.9.9 getSSL

Online command that lists the configured SSL attributes.

6.9.9.1 Description

This command lists the configured SSL attributes for the specified component listener.

6.9.9.2 Syntax

```
getSSL('instName',#'compName',#'compType',#'listener')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
listener	Specifies the name of the component listener.

6.9.9.3 Example

The following command shows the SSL attributes configured for Oracle HTTP Server instance ohs1, in application server instance inst1, for listener sslport1:

```
wls:/mydomain/serverConfig>#getSSL('inst1',#'ohs1',#'ohs',#'sslport1')
```

6.9.10 getWalletObject

Online command that displays information about a certificate or other object in an Oracle wallet.

6.9.10.1 Description

This command displays a specific certificate signing request, certificate or trusted certificate present in an Oracle wallet for the specified component instance. The wallet object is indicated by its index number, as given by the `listWalletObjects` command. For certificates or trusted certificates, it shows the certificate details including DN, key size, algorithm and other data. For certificate signing requests, it shows the subject DN, key size and algorithm.

6.9.10.2 Syntax

```
getWalletObject('instName',#'compName',#'compType',#'walletName',#'password',#'type',#'index')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file.
password	Specifies the password of the wallet.
type	Specifies the type of wallet object to be exported. Valid values are 'CertificateRequest', 'Certificate', and 'TrustedCertificate'.

Argument	Definition
index	Specifies the index number of the wallet object as returned by the <code>listWalletObjects</code> command.

6.9.10.3 Examples

The following command shows certificate signing request details for the object with index 0 present in `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#getKeyStoreObject('inst1','#ohs1',#
'ohs','wallet1','password','#CertificateRequest','#0')
```

The following command shows certificate details for the object with index 0 present in `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#getKeyStoreObject('inst1','#ohs1',#
'ohs','wallet1','password','#Certificate','#0')
```

The following command shows trusted certificate details for the object with index 0, present in `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#getKeyStoreObject('inst1','#ohs1',#
'ohs','wallet1','password','#TrustedCertificate','#0')
```

6.9.11 importWallet

Online command that imports an Oracle wallet from a file.

6.9.11.1 Description

This command imports an Oracle wallet from a file to the specified component instance for manageability. If the wallet being imported is an auto-login wallet, the file path must point to `cwallet.sso`; if the wallet is password-protected, it must point to `ewallet.p12`. The wallet name must be unique for the component instance.

6.9.11.2 Syntax

```
importWallet('instName', 'compName', 'compType', 'walletName', 'password',
'filePath')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet being imported. The name must be unique for the component instance.
password	Specifies the password of the wallet.
filePath	Specifies the absolute path of the wallet file being imported.

6.9.11.3 Examples

The following command imports the auto-login wallet file `/tmp/cwallet.sso` as `wallet1` into Oracle HTTP Server instance `ohs1`. Subsequently, the wallet is managed with the name `wallet1`. No password is passed since it is an auto-login wallet:

```
wls:/mydomain/serverConfig>#importWallet('inst1',#'ohs1',#'ohs',#'wallet1',#',#
'/tmp/cwallet.sso')
```

The following command imports password-protected wallet `/tmp/ewallet.p12` as `wallet2` into Oracle HTTP Server instance `ohs1`. Subsequently, the wallet is managed with the name `wallet2`. The wallet password is passed as a parameter:

```
wls:/mydomain/serverConfig>#importWallet('inst1',#'ohs1',#'ohs',#'wallet2',#
'password',#'tmp/ewallet.p12')
```

6.9.12 importWalletObject

Online command that imports a certificate or other object into an Oracle wallet.

6.9.12.1 Description

This command imports a certificate, trusted certificate or certificate chain into an Oracle wallet for the specified component instance. When importing a certificate, use the same wallet file from which the certificate signing request was generated.

6.9.12.2 Syntax

```
importWalletObject('instName', 'compName', 'compType', 'walletName', 'password',
'type', 'filePath')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file.
password	Specifies the password of the wallet.
type	Specifies the type of wallet object to be imported. Valid values are 'Certificate', 'TrustedCertificate' and 'TrustedChain'.
filePath	Specifies the absolute path of the file containing the wallet object.

6.9.12.3 Examples

The following command imports a certificate chain in PKCS#7 format from file `chain.txt` into `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#importWalletObject('inst1',#'ohs1',#'ohs',#'wallet1',#
'password',#'TrustedChain',#'tmp/chain.txt')
```

The following command imports a certificate from file `cert.txt` into `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#>#importWalletObject('inst1',#'ohs1',#'ohs',#'wallet1',#
'password',#'Certificate',#'tmp/cert.txt')
```

The following command imports a trusted certificate from file `trust.txt` into `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#importWalletObject('inst1',#'ohs1',#'ohs',#'wallet1',#
'password',#'TrustedCertificate',#'tmp/trust.txt')
```

6.9.13 listWalletObjects

Online command that lists all objects in an Oracle wallet.

6.9.13.1 Description

This command lists all certificate signing requests, certificates, or trusted certificates present in an Oracle wallet for the specified component instance.

6.9.13.2 Syntax

```
listWalletObjects('instName',# 'compName',# 'compType',# 'walletName',#password',# 'type')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file.
password	Specifies the password of the wallet.
type	Specifies the type of wallet object to be listed. Valid values are 'CertificateRequest', 'Certificate', and 'TrustedCertificate'.

6.9.13.3 Examples

The following command lists all certificate signing requests in wallet1, for Oracle HTTP Server instance ohs1, in application server instance inst1:

```
wls:/mydomain/serverConfig>#listWalletObjects('inst1',# 'ohs1',# 'ohs', 'wallet1', 'password',# 'CertificateRequest')
```

The following command lists all certificates in wallet1, for Oracle HTTP Server instance ohs1, in application server instance inst1:

```
wls:/mydomain/serverConfig>#listWalletObjects('inst1',# 'ohs1',# 'ohs', 'wallet1', 'password',# 'Certificate')
```

The following command lists all trusted certificates in wallet1, for Oracle HTTP Server instance ohs1, in application server instance inst1:

```
wls:/mydomain/serverConfig>#listWalletObjects('inst1',# 'ohs1',# 'ohs', 'wallet1', 'password',# 'TrustedCertificate')
```

6.9.14 listWallets

Online command that lists all wallets configured for a component instance.

6.9.14.1 Description

This command displays all the wallets configured for the specified component instance, and identifies the auto-login wallets.

6.9.14.2 Syntax

```
listWallets('instName',# 'compName',# 'compType')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance
compType	Specifies the type of component. Valid value is 'ohs'.

6.9.14.3 Example

The following command lists all wallets for Oracle HTTP Server instance ohs1 in application server instance inst1:

```
wls:/mydomain/serverConfig>#listWallets('inst1','#ohs1','#ohs')
```

6.9.15 removeWalletObject

Online command that removes a certificate or other object from an Oracle wallet.

6.9.15.1 Description

This command removes a certificate signing request, certificate, trusted certificate or all trusted certificates from an Oracle wallet for the specified component instance. DN is used to indicate the object to be removed.

6.9.15.2 Syntax

```
removeWalletObject('instName', 'compName', 'compType', 'walletName', 'password', 'type', 'DN')
```

Argument	Definition
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid value is 'ohs'.
walletName	Specifies the name of the wallet file.
password	Specifies the password of the wallet.
type	Specifies the type of the keystore object to be removed. Valid values are 'CertificateRequest', 'Certificate', 'TrustedCertificate' or 'TrustedAll'.
DN	Specifies the Distinguished Name of the wallet object to be removed.

6.9.15.3 Examples

The following command removes all trusted certificates from wallet1, for Oracle HTTP Server instance ohs1, in application server instance inst1. It is not necessary to provide a DN, so you pass null (denoted by None) for the DN parameter:

```
wls:/mydomain/serverConfig>#removeWalletObject('inst1','#ohs1','#ohs','wallet1','#password','#TrustedAll',None)
```

The following command removes a certificate signing request indicated by DN cn=www.acme.com from wallet1, for Oracle HTTP Server instance ohs1, in application server instance inst1:

```
wls:/mydomain/serverConfig>#removeWalletObject('inst1','#ohs1','#ohs','wallet1','#password','#CertificateRequest','cn=www.acme.com')
```


The following command removes a certificate indicated by DN `cn=www.acme.com` from `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#removeWalletObject('inst1',#'ohs1',#'ohs','wallet1',#
'password',#'Certificate','cn=www.acme.com')
```

The following command removes a trusted certificate indicated by DN `cn=www.acme.com` from `wallet1`, for Oracle HTTP Server instance `ohs1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig>#removeWalletObject('inst1',#'ohs1',#'ohs','wallet1',#
'password',#'TrustedCertificate','cn=www.acme.com')
```

6.9.16 Properties Files for SSL

SSL configuration employs certain properties files for use with the `WLST configureSSL` command. The files contain parameters to specify the desired SSL configuration, such as authentication type, cipher values, and SSL version.

You can use descriptive names if you need to manage multiple properties files for different components. For example, you could have properties files named `ohs-ssl-properties.prop` or `ovd-ssl-properties.prop`.

6.9.16.1 Structure of Properties Files

All the SSL properties files have a consistent structure.

[Table 6–3](#) provides details about the key-value structure and usage of these files.

Table 6–3 Parameters in Properties File

Key	Mandatory?	Allowed Values for Oracle HTTP Server	Usage
SSLEnabled	No	true false	Either value
Ciphers	No	See "SSLCipherSuite" in <i>Administering Oracle HTTP Server</i> .	One or more comma separated values
SSLVersions	No	See "SSLProtocol" in <i>Administering Oracle HTTP Server</i> .	One or more comma separated values
CertValidation	No	none crl	Either value
CertValidation# Path	No	file://crl_file_path dir://crl_dir_path	Path of the CRL file, or directory containing CRL files
KeyStore	No	Valid wallet name	
TrustStore	No	N/A	
AuthenticationType	No	None Server Optional Mutual	Any one value

[Table 6–4](#) shows the default values:

Table 6–4 Default Values of Parameters

Key	Default Value for Oracle HTTP Server
SSLEnabled	true
Ciphers	null
SSLVersions	null
CertValidation	none
CertValidation# Path	null
KeyStore	default
TrustStore	-
Authentication# Type	Server

Note:

- At least one DH_anon cipher must be used in SSL no-auth mode. For all other modes, at least one RSA cipher must be used.
 - The value of the KeyStore parameter must be specified when configuring SSL in server-auth, mutual-auth, or optional client auth.
 - If only AES ciphers have been specified, the SSLVersions parameter must contain TLSv1 or nzos_Version_1_0.
 - If you are doing CRL-based validation, the value of the CertValidation parameter should be `crl` and the value of the CertValidationPath parameter should point to the CRL file/directory.
-
-

6.9.16.2 Examples of Properties Files

Some examples demonstrating the use of the properties files follow.

Example 1: Basic Properties File

```
SSLEnabled=true
AuthenticationType=None
CertValidation=none
```

This properties file specifies no authentication mode, and default values will be used during SSL configuration for ciphers and SSL version. Keystore and truststore properties are not specified since the authentication type is `None`. For other authentication types, keystore must be specified.

Example 2: Basic Properties File

```
SSLEnabled=
AuthenticationType=None
CertValidation=none
```

This properties file is exactly the same as above, except that `SSLEnabled` is explicitly specified without any value. This is the same as not specifying the key at all. In both cases, the default value will be used.

Therefore, all the following three settings have the same meaning:

- The setting:

```
SSLEnabled=true
```

Here the value `true` is explicitly specified.

- The setting:

```
SSLEnabled=
```

Since no value is mentioned here, the default value of `SSLEnabled` (`true`) is used.

- The key `SSLEnabled` is not present in the properties file.

Since the key is not present, its default value (`true`) is used.

Example 3: Properties File with Version for OHS

```
SSLEnabled=true
AuthenticationType=Mutual
SSLVersion=nzos_Version_1_0
CertValidation=crl
CertValidationPath=file:///tmp/file.crl
KeyStore=ohs1
```

This properties file has:

- Default values for ciphers
- Keystore
- SSL version v1
- CRL validation turned on
- Mutual Authentication mode

Managing Keystores, Wallets, and Certificates

This chapter explains how to use Oracle Fusion Middleware security features to administer keystores, keys, and certificates.

This chapter contains these sections:

- [Key and Certificate Storage in Oracle Fusion Middleware](#)
- [Command-Line Interface for Keystores and Wallets](#)
- [Keystore Management](#)
- [Wallet Management](#)

7.1 Key and Certificate Storage in Oracle Fusion Middleware

Keys and certificates are used to digitally sign and verify data and achieve authentication, integrity, and privacy in network communications.

Private keys, digital certificates, and trusted CA certificates are stored in keystores. This section describes the keystores available in Oracle Fusion Middleware and contains these topics:

- [Types of Keystores](#)
- [Keystore Management Tools](#)

7.1.1 Types of Keystores

Oracle Fusion Middleware provides various types of keystores for keys and certificates as shown in [Table 7-1](#):

Table 7-1 Keystore Types in Oracle Fusion Middleware

Keystore Type	Description	Protection Mechanism
Oracle Wallet	Oracle Wallet	Password or auto-login
JKS	Java Keystore	Password
KSS	OPSS Keystore Service	Password or policy

7.1.1.1 About Oracle Wallet

An Oracle wallet is a container that stores your credentials, such as certificates, trusted certificates, certificate requests, and private keys. You can store Oracle wallets on the file system or in LDAP directories such as Oracle Internet Directory. Oracle wallets can be auto-login or password-protected wallets.

When creating a wallet, you can:

- pre-populate it with a self-signed certificate; such a wallet is called a test wallet and is typically used in development and testing phases.
- create a certificate request, so that you can request a signed certificate back from a Certificate Authority (CA). Once the CA sends the certificate back, it is imported into the wallet; such a wallet is called a third-party wallet.

Either the test wallet or the third-party wallet may be password-protected, or may be configured to not require a password, in which case it is called an auto-login wallet.

Oracle Wallets are used for Oracle HTTP Server.

7.1.1.2 About the JKS Keystore

A JKS keystore is the default JDK implementation of Java keystores. Java EE applications can use the JKS-based keystore and truststore.

7.1.1.3 About the Keystore Service (KSS) Keystore

Introduced in 11g Release 1 (11.1.1.6.0), the OPSS Keystore Service enables you to manage keys and certificates for SSL, message security, encryption, and related tasks.

The Keystore Service offers several advantages including policy-based protection and centralized management of keystores and truststores, expiring certificates, and other key material.

In Oracle Fusion Middleware 12c (12.1.2), Oracle WebLogic Server:

- uses the Keystore Service out-of-the-box
- uses JKS by default in upgraded environments.

7.1.2 Keystore Management Tools

Oracle Fusion Middleware provides these options for keystore operations:

- WLST, a command-line interface for JKS keystores and wallets
- orapki, a command-line tool for wallets
- Fusion Middleware Control, a graphical user interface

See Also: For details about using keytool, see *Administering Security for Oracle WebLogic Server*.

About Importing DER-encoded Certificates

You cannot use Fusion Middleware Control or the WLST command-line tool to import DER-encoded certificates or trusted certificates into an Oracle wallet. Use the `orapki` command-line tool instead.

Using a Keystore Not Created with WLST or Fusion Middleware Control

If an Oracle wallet was created with tools such as `orapki`, it must be imported prior to use. Specifically for Oracle HTTP Server, if a wallet was created using `orapki`, in order to view or manage it in Fusion Middleware Control you must first import it with either Fusion Middleware Control or the WLST `importWallet` command. For details, see [Section 7.4.4.9](#) and [Section 7.4.4.10](#).

Copying Keystores to File System Not Supported

Creating, renaming, or copying keystores directly to any directory on the file system is not supported. Any existing pre-11g keystore or wallet that you wish to use must be imported using either Fusion Middleware Control or the WLST utility.

Additional Information

Details about the tools are provided in these sections:

- [Command-Line Interface for Keystores and Wallets](#)
- [Wallet Management](#)
- [Appendix G, "orapki"](#)

7.2 Command-Line Interface for Keystores and Wallets

Oracle Fusion Middleware provides a set of `wlst` scripts to create and manage keystores and Oracle wallets, and to manipulate their stored objects.

How to Launch the WLST Command-Line Interface

When running SSL WLST commands, you must invoke the `WLST` script from the Oracle Common home. See [Section 2.4.2](#) for more information.

Note: All SSL-related WLST commands require you to launch the script from the above-mentioned location only.

This brings up the WLST shell. Connect to a running Oracle WebLogic Server instance by specifying the user name, password, and connect URL. After connecting, you are now ready to run SSL-related WLST commands as explained in the subsequent sections.

7.3 Keystore Management

For details about Keystore Service (KSS) keystore management, see "Keystore Management with the Keystore Service" in *Securing Applications with Oracle Platform Security Services*.

7.4 Wallet Management

This section contains the following topics:

- [About Wallets and Certificates](#)
- [Accessing the Wallet Management Page in Fusion Middleware Control](#)
- [Managing the Wallet Life Cycle](#)
- [Common Wallet Operations](#)
- [Managing the Certificate Life Cycle](#)
- [Accessing the Certificate Management Page for Wallets in Fusion Middleware Control](#)
- [Common Certificate Operations](#)
- [Wallet and Certificate Maintenance](#)

7.4.1 About Wallets and Certificates

This section contains the following topics:

- [Password-Protected and Autologin Wallets](#)
- [Self-Signed and Third-Party Wallets](#)
- [Sharing Wallets Across Instances](#)
- [Wallet Naming Conventions](#)

7.4.1.1 Password-Protected and Autologin Wallets

You can create two types of wallets:

- Auto-login wallet

This is an obfuscated form of a PKCS#12 wallet that provides PKI-based access to services and applications without requiring a password at runtime. You can also add to, modify, or delete the wallet without needing a password. File system permissions provide the necessary security for auto-login wallets.

Note: In previous releases, you could create a wallet with a password and then enable auto-login to create an obfuscated wallet. With 12c (12.1.2), auto-login wallets are created without a password. When using such a wallet, you do not need to specify a password.

If using an auto-login wallet without a password, specify a null password ("") in the `ldapbind` command.

Older type of wallets (such as Release 10g wallets) will continue to work as they did earlier.

- Password-protected wallet

As the name suggests, this type of wallet is protected by a password. Any addition, modification, or deletion to the wallet content requires a password.

Every time a password-protected wallet is created, an auto-login wallet is automatically generated. However, this auto-login wallet is different from the user-created auto-login wallet described in the previous bullet. While the user-created wallet can even be updated at configuration time without a password, an automatically generated auto-login wallet is a read-only wallet that does not allow direct updates. Modifications to the wallet must occur through the password protected file (by providing a password), at which time the auto-login wallet is regenerated.

The purpose of this system-generated auto-login wallet is to provide PKI-based access to services and applications without requiring a password at runtime, while still requiring a password at configuration time.

Note: Wallets configured for Oracle Internet Directory must have auto-login enabled.

7.4.1.2 Self-Signed and Third-Party Wallets

Self-signed wallets contain certificates for which the issuer is the same as the subject. These wallets are typically created for use within an intranet environment where trust is not a high priority. Each self-signed wallet has its own unique issuer; hence, in an

environment with multiple components and wallets, the trust management tasks increase n-fold.

When created through Fusion Middleware Control, a self-signed wallet is valid for five years.

Third-party wallets contain certificates that are issued by well known CA's. The functionality and security remain the same as for self-signed wallets, but the use of third-party certificates provides added trust because the issuers are well known, so they are already trusted by most clients.

Difference Between Self-Signed and Third-Party Wallets

From a functional and security perspective, a self-signed certificate is comparable to one issued by a third party. The only difference is that a self-signed certificate is not trusted.

7.4.1.3 Sharing Wallets Across Instances

Oracle recommends that you do not share wallets between component instances, since each wallet represents a unique identity.

The exception to this is an environment with a cluster of component instances, in which case wallet sharing would be an acceptable practice.

Note that no management tools or interfaces are available to facilitate wallet sharing. However, you can export a wallet from one instance and import it into another instance. See [Section 7.4.4](#) for details of wallet export and import.

7.4.1.4 Wallet Naming Conventions

Follow these naming conventions for your Oracle wallets:

- Do not use a name longer than 256 characters.
- Do not use any of the following characters in a wallet name:


```
|#;#,#!#@###$#(#)#<#>#/#\#"#'#`#~#{#}#[#]#=#+#&#^#space#tab
```

Note: Observe this rule even your operating system supports the character.

- Do not use non-ascii characters in a wallet name.
- Additionally, follow the operating system-specific rules for directory and file names

Due to the way data is handled in an LDAP directory such as Oracle Internet Directory, wallet names are not case-sensitive.

Thus, it is recommended that you use case-insensitive wallet names (preferably, using all lower case letters). For example, if you have created a wallet named UPPER, do not create another wallet named upper; doing so could cause confusion during wallet management operations.

7.4.2 Accessing the Wallet Management Page in Fusion Middleware Control

An Oracle wallet is associated with the component where it is utilized. To locate a component instance:

1. Log into Fusion Middleware Control using administrator credentials.
2. Select the domain of interest.

Note: You can use Setup to discover a specific Oracle WebLogic Server domain to work with.

3. From the navigation pane, locate the instance (for example, an OHS instance) that will use the wallet. Click on the instance.

The component type now appears on the upper left of the page adjacent to the Farm drop-down.

4. Select the component type drop-down (for example, Oracle HTTP Server).

If the component is not started, start it by right-clicking to open the component menu, press **Control**, then **Start Up**.

5. Navigate to **Security**, then **Wallets**.

6. The Wallets page appears.

On the Wallets page, you can:

- Create a wallet.
- Delete a wallet.
- Import a wallet.
- Export a wallet.

7.4.3 Managing the Wallet Life Cycle

Typical life cycle events for an Oracle wallet are as follows:

- The wallet is created. Wallets can be created directly, or by importing a wallet file from the file system.
- The list of available wallets are viewed and specific wallets selected for update.
- Wallets are updated or deleted. Update operations for password-protected wallets require that the wallet password be entered.
- The wallet password can be changed for password-protected wallets.
- The wallet can be deleted.
- Wallets can be exported and imported.

7.4.4 Common Wallet Operations

This section describes the steps required to perform a range of wallet management functions, including:

- [Creating a Wallet Using Fusion Middleware Control](#)
- [Creating a Wallet Using WLST](#)
- [Creating a Self-Signed Wallet Using Fusion Middleware Control](#)
- [Creating a Self-Signed Wallet Using WLST](#)
- [Changing a Self-Signed Wallet to a Third-Party Wallet Using Fusion Middleware Control](#)

- [Changing a Self-Signed Wallet to a Third-Party Wallet Using WLST](#)
- [Exporting a Wallet Using Fusion Middleware Control](#)
- [Exporting a Wallet Using WLST](#)
- [Importing a Wallet Using Fusion Middleware Control](#)
- [Importing a Wallet Using WLST](#)
- [Deleting a Wallet Using Fusion Middleware Control](#)
- [Deleting a Wallet Using WLST](#)

7.4.4.1 Creating a Wallet Using Fusion Middleware Control

Take these steps to a wallet:

1. Navigate to the Wallets page for your component instance. See [Section 7.4.2, "Accessing the Wallet Management Page in Fusion Middleware Control."](#)
2. Click **Create**.
3. The Create Wallet page appears.
4. Enter a wallet name.
5. Check or uncheck the **Autologin** box, depending on whether your wallet will be an auto-login wallet. The default is an auto-login wallet.

See [Section 7.4.1.1, "Password-Protected and Autologin Wallets"](#) for details.

Note: Wallets configured for Oracle Internet Directory must have auto-login enabled.

Oracle Internet Directory Page Refreshed Feb 6, 2009 3:08:50 PM PST

Wallets > Create Wallet

Create Wallet

The wallet name should be unique for a given component. The wallet type can be auto-login or password-protected. Passwords, if specified, have a minimum length of eight characters, and contain alphabetic characters combined with numeric or special characters. Auto-login wallet is an obfuscated form of PKCS#12 wallet that provides PKI-based access to services and applications without requiring a password at runtime. Auto-login wallet don't need a password to modify, or delete the wallet. File system permissions provide the necessary security for Auto-login wallets.

Details

* Wallet Name

Auto-login

* Wallet Password

* Confirm Password

6. Click **Submit**.
7. At this point, you must choose whether to add a certificate request (CR) at this time. If you choose not to do so, you can always add the CR later; see [Section 7.4.7.1, "Adding a Certificate Request Using Fusion Middleware Control."](#)

In this example, we choose to add a CR:

Oracle Internet Directory Page Refreshed Feb 6, 2009 3:11:42 PM PST

Wallets > Create Wallet : Add Certificate Request

Create Wallet : Add Certificate Request

To generate a certificate request (CR), enter the details of the CR. Click OK to generate a CR that you can cut and paste or export to a file. You typically do this to get a certificate signed by a Certificate Authority.

* Common Name

Organizational Unit

Organization

City

State

Country

Key Size

Note: The common name entered here should match the host name of the Oracle HTTP Server to which clients will connect; this helps to prevent problems of the type mentioned in [Section 7.4.8.2](#).

8. Click **Finish**.
9. There are two options for the CR:
 - Copy and paste the Base64-encoded certificate request from the text box to a file
 - Export it directly to a file with the **Export Certificate Request** button.
10. A message appears confirming the wallet creation.

7.4.4.2 Creating a Wallet Using WLST

Assuming the instance name is `inst1`, use this command to create a wallet:

```
createWallet('inst1',#'ohs1',#'ohs',#'ohs2',#'password')
```

where `ohs2` is the wallet name and `password` is the password for this wallet. If an auto-login wallet needs to be created, the password should be specified as "" (that is, no text between the quotes).

See Also: [Section 6.9.5, "createWallet"](#).

7.4.4.3 Creating a Self-Signed Wallet Using Fusion Middleware Control

Take these steps to create a self-signed wallet:

See Also: [Section 7.4.1.2, "Self-Signed and Third-Party Wallets"](#)

1. Navigate to the Wallets page for your component instance. See [Section 7.4.2, "Accessing the Wallet Management Page in Fusion Middleware Control"](#).
2. Click **Create Self-Signed Wallet**.
3. On the Self-Signed Wallet page, enter data to create the wallet. This includes:
 - The wallet name
 - Whether this is an auto-login wallet

See Also: [Section 7.4.1.1, "Password-Protected and Autologin Wallets"](#)

- The DN information
- The key size

Oracle Internet Directory Page Refreshed Feb 6, 2009 3:13:56 PM PST

Wallets > Create Self-Signed Wallet

Create Self-Signed Wallet

A self signed wallet is not signed by a well known CA. A self-signed wallet is not recommended in a production environment. The wallet name should be unique for a given component. The wallet type can be auto-login or password-protected. Passwords, if specified, have a minimum length of eight characters, and contain alphabetic characters combined with numeric or special characters. Auto-login wallet is an obfuscated form of PKCS#12 wallet that provides PKI-based access to services and applications without requiring a password at runtime. Auto-login wallet don't need a password to modify, or delete the wallet. File system permissions provide the necessary security for Auto-login wallets.

Self-Signed Wallet Details

* Wallet Name: selfsigned

Auto-login

Wallet Password: _____

Confirm Password: _____

Add Self-Signed Certificate

Add a self-signed certificate that becomes part of the wallet.

* Common Name: ldap.acme.com

Organizational Unit: FOR TESTING ONLY

Organization: _____

City: _____

State: _____

Country: United States

Key Size: 1024

4. Click **Submit**.
5. A confirmation message is displayed and the new wallet appears in the list of wallets.

Oracle Internet Directory Page Refreshed Feb 6, 2009 3:16:12 PM PST

Confirmation
Self-signed wallet selfsigned successfully created

Wallets
A Wallet is a Keystore that stores X.509 certificates and private keys in industry-standard, PKCS #12 format. To create a wallet, click Create. To create a wallet with a self-signed certificate, click Create Self-Signed Wallet. To manage the contents of a wallet, select a wallet and click Manage.

Name	Auto-login
oid2	
selfsigned	<input checked="" type="checkbox"/>

Note: Wallets configured for Oracle Internet Directory must have auto-login enabled.

7.4.4.4 Creating a Self-Signed Wallet Using WLST

Assuming the instance name is `inst1`, use these commands to create a self-signed wallet:

```
createWallet('inst1',#'ohs1',#'ohs',#'ohs2',#'password')
addSelfSignedCertificate('inst1',#'ohs1',#'ohs',#'ohs2',#'password',#'subject_dn',#'key_size')
```

where `ohs2` is the wallet name, `subject_dn` is the distinguished name of the self-signed certificate, `key_size` is the key size in bits and `password` is the password for this wallet. If an auto-login wallet needs to be created, the password should be specified as "" (that is, with no text between the quotes).

See Also:

- [Section 6.9.5, "createWallet"](#)
- [Section 6.9.2, "addSelfSignedCertificate"](#)

Note: Wallets configured for Oracle Internet Directory must have auto-login enabled.

7.4.4.5 Changing a Self-Signed Wallet to a Third-Party Wallet Using Fusion Middleware Control

For steps to convert a self-signed wallet into a third-party wallet, see [Section 7.4.8.3, "Changing a Self-Signed Wallet to a Third-Party Wallet."](#)

7.4.4.6 Changing a Self-Signed Wallet to a Third-Party Wallet Using WLST

For steps to convert a self-signed wallet into a third-party wallet, see [Section 7.4.8.3, "Changing a Self-Signed Wallet to a Third-Party Wallet."](#)

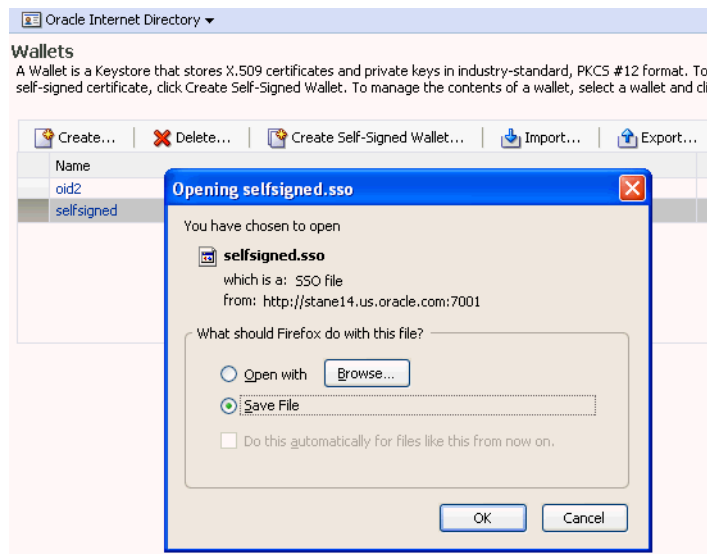
7.4.4.7 Exporting a Wallet Using Fusion Middleware Control

Take these steps to export a wallet:

1. Navigate to the Wallets page for your component instance. See [Section 7.4.2, "Accessing the Wallet Management Page in Fusion Middleware Control."](#)
2. Select the row corresponding to the wallet of interest.

Note: Do not click on the wallet name itself; this opens the wallet for certificate management operations.

3. Click **Export**.
4. The Export Wallet page appears.
5. Enter the filename and the location where the wallet is to be exported.
6. Click **OK**.



7.4.4.8 Exporting a Wallet Using WLST

Assuming the instance name is `inst1`, use this command to export a wallet:

```
exportWallet('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#'/tmp')
```

where `password` is the password for this wallet (specify `"` as password for auto-login wallet).

If it is an auto-login wallet, this command will export the wallet into a file named `cwallet.sso` under the directory `/tmp`. If it is a password-protected wallet, there will be two files created under `/tmp`, namely `ewallet.p12` and `cwallet.sso`.

See Also: [Section 6.9.7, "exportWallet"](#).

7.4.4.9 Importing a Wallet Using Fusion Middleware Control

Take these steps to import a wallet:

1. Navigate to the Wallets page for your component instance. See [Section 7.4.2, "Accessing the Wallet Management Page in Fusion Middleware Control"](#).
2. Click **Import**.
3. The Import Wallet page appears.
4. If this is an auto-login wallet, check the box and enter the wallet name. No password is required.

Oracle Internet Directory Page Refreshed Feb 6, 2009 3:18:30 PM PST

Wallets > Import Wallet

Import Wallet OK Cancel

Click "Browse" to select the wallet and import it for the selected component. The wallet name should be unique for the component. Password-protected wallet files usually have a ".p12" extension and auto-login wallets have a ".sso" extension. You import a wallet to use for a component, for example with SSL operations.

File: C:\cwallet.sso Browse...

Auto-login

* Wallet Name:

Wallet Password:

5. If this is not an auto-login wallet, uncheck the auto-login box. Specify both the wallet name and password.

Oracle Internet Directory Page Refreshed Feb 6, 2009 3:18:30 PM PST

Wallets > Import Wallet

Import Wallet OK Cancel

Click "Browse" to select the wallet and import it for the selected component. The wallet name should be unique for the component. Password-protected wallet files usually have a ".p12" extension and auto-login wallets have a ".sso" extension. You import a wallet to use for a component, for example with SSL operations.

File: C:\ewallet.p12 Browse...

Auto-login

* Wallet Name:

* Wallet Password:

6. Click **OK**. The wallet is imported into the repository.

7.4.4.10 Importing a Wallet Using WLST

Assuming the instance name is `inst1`, use this command to import a wallet:

```
importWallet('inst1',#'ohs1',#'ohs',#'ohs5',#'password',#'/tmp/ewallet.p12')
```

where `password` is the password of the wallet being imported and `/tmp/ewallet.p12` contains the wallet file (if there are two files `ewallet.p12` and `cwallet.sso`, point to `ewallet.p12`). Point to `cwallet.sso` only if it is an auto-login wallet - in this case, the password should be specified as "".

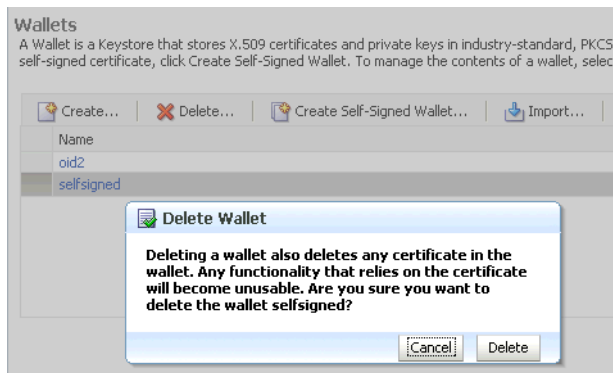
See Also: [Section 6.9.11, "importWallet"](#).

7.4.4.11 Deleting a Wallet Using Fusion Middleware Control

Take these steps to delete a wallet:

1. Navigate to the Wallets page for your component instance. See [Section 7.4.2, "Accessing the Wallet Management Page in Fusion Middleware Control."](#)

2. Select the row corresponding to the wallet of interest.
3. Click **Delete**.



4. The wallet is deleted and no longer appears on the list of wallets.

7.4.4.12 Deleting a Wallet Using WLST

Assuming the instance name is `inst1`, use this command to delete a wallet:

```
deleteWallet('inst1',#'ohs1',#'ohs',#'selfsigned')
```

See Also: [Section 6.9.6, "deleteWallet"](#).

7.4.5 Managing the Certificate Life Cycle

The complete certificate life cycle, starting from wallet creation, includes these actions:

1. Create an empty wallet (that is, a wallet that does not contain a certificate request).
2. Add a certificate request to the wallet.
3. Export the certificate request.
4. Use the certificate request to obtain the corresponding certificate.
5. Import trusted certificates.
6. Import the certificate.

These steps are needed to generate a wallet with a third-party trusted certificate. For details about this task, see [Section 7.4.7.9, "Converting a Self-Signed Certificate into a Third-Party Certificate Using Fusion Middleware Control."](#)

See Also: [Section 7.4.6, "Accessing the Certificate Management Page for Wallets in Fusion Middleware Control"](#)

7.4.6 Accessing the Certificate Management Page for Wallets in Fusion Middleware Control

An Oracle wallet is associated with the component where it is utilized.

To locate a component instance:

1. Log into Fusion Middleware Control using administrator credentials.
2. Select the domain of interest.

Note: You can use Setup to discover a specific Oracle WebLogic Server domain to work with.

3. Use the navigation pane to locate the instance (for example, an Oracle HTTP Server instance) that will use the wallet. *Note:* Oracle HTTP Server must be running so that subsequent steps will work.
 - Use the navigation pane to locate the instance (for example, an Oracle HTTP Server instance) that will use the wallet. *Note:* Oracle HTTP Server must be running so that subsequent steps will work.

After locating your component instance, there are two ways you can access a wallet's certificate management page in Fusion Middleware Control:

- Go to the *Wallets* page, select the line for the wallet of interest and click **Manage**.
- Go to the *Wallets* page, locate the wallet of interest, and click on the wallet name.

On the Certificate Management page, you can:

- Add a certificate request.
- Export a certificate request, a certificate, or a trusted certificate.
- Import a certificate or a trusted certificate.
- Delete a certificate request, a certificate, or a trusted certificate.

7.4.7 Common Certificate Operations

This section describes the following common certificate operations:

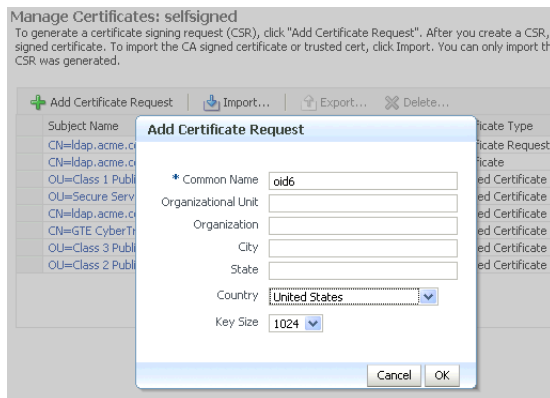
- [Adding a Certificate Request Using Fusion Middleware Control](#)
- [Adding a Certificate Request Using WLST](#)
- [Exporting a Certificate, Certificate Request, or a Trusted Certificate Using Fusion Middleware Control](#)
- [Exporting a Certificate, Certificate Request, or a Trusted Certificate Using WLST](#)
- [Importing a Certificate or a Trusted Certificate Using Fusion Middleware Control](#)
- [Importing a Certificate or a Trusted Certificate Using WLST](#)
- [Deleting a Certificate Request, a Certificate, or a Trusted Certificate Using Fusion Middleware Control](#)
- [Deleting a Certificate Request, a Certificate, or a Trusted Certificate Using WLST](#)
- [Converting a Self-Signed Certificate into a Third-Party Certificate Using Fusion Middleware Control](#)
- [Converting a Self-Signed Certificate into a Third-Party Certificate Using WLST](#)

7.4.7.1 Adding a Certificate Request Using Fusion Middleware Control

It is possible to add a certificate request at the time you create the wallet, but if it was not done at that time, you can do so using the following steps:

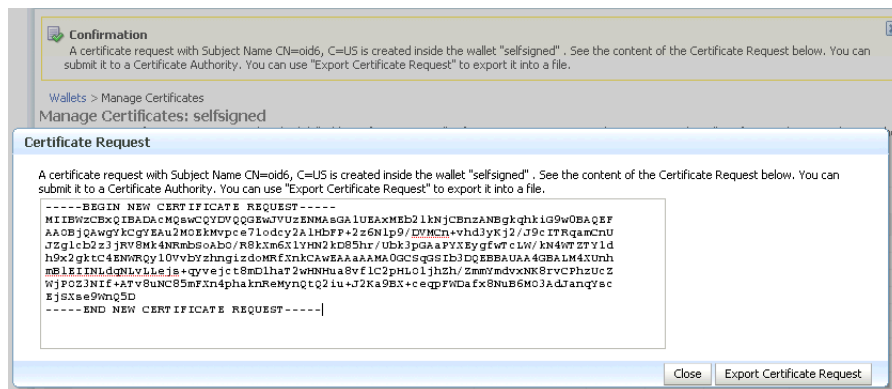
1. Navigate to the Certificate Management page. See [Section 7.4.6, "Accessing the Certificate Management Page for Wallets in Fusion Middleware Control."](#)
2. Click **Add Certificate Request**.

- A dialog box appears where you enter the CRs DN values:



Fields marked with an asterisk (*) are required. *Note:* The common name must be the host name that clients will use to access the component.

- Click **OK**.
- The new CR is generated and a dialog box appears with the CR in the text box. You can either:
 - Copy and paste the Base64-encoded certificate request to a file.
 - Export it directly to a file with the **Export Certificate Request** button.



7.4.7.2 Adding a Certificate Request Using WLST

Assuming the instance name is `inst1`, use this command to add a certificate request for a wallet:

```
addCertificateRequest('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#'subject_
dn',#'key_size')
```

where `password` is the password for this wallet, `subject_dn` is the distinguished name by which the certificate request is generated and `key_size` is the key size in bits.

See Also: [Section 6.9.1](#)

7.4.7.3 Exporting a Certificate, Certificate Request, or a Trusted Certificate Using Fusion Middleware Control

Take these steps to export a certificate, a certificate request (CR), or a trusted certificate:

1. Navigate to the Certificate Management page. See [Section 7.4.6, "Accessing the Certificate Management Page for Wallets in Fusion Middleware Control."](#)
2. Select the certificate, CR, or trusted certificate and click **Export**.
3. A dialog box appears with the certificate, CR, or trusted certificate in the text box. You can either:
 - Copy and paste the Base64-encoded certificate to a file.
 - Export it directly to a file with the **Export Certificate** or **Export Trusted Certificate** button.

7.4.7.4 Exporting a Certificate, Certificate Request, or a Trusted Certificate Using WLST

Assuming the instance name is `inst1`, use this command to export a certificate request:

```
exportWalletObject('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'CertificateRequest',#'/tmp',#'subject_dn')
```

where `password` is the password for this wallet, `/tmp` is the path under which the certificate request is exported in BASE64 format in the file `base64.txt`, and `subject_dn` is the distinguished name of the certificate request that is exported.

To export a certificate or trusted certificate, replace `CertificateRequest` in the above command with `Certificate` or `TrustedCertificate`.

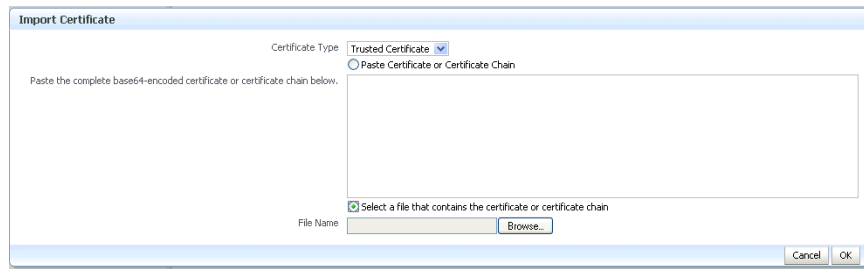
See Also: [Section 6.9.8](#)

7.4.7.5 Importing a Certificate or a Trusted Certificate Using Fusion Middleware Control

Note: You cannot use Fusion Middleware Control to import DER-encoded certificates or trusted certificates into an Oracle wallet. Use `orapki` command-line tool instead.

Take these steps to import a certificate or a trusted certificate:

1. Navigate to the Certificate Management page. See [Section 7.4.6, "Accessing the Certificate Management Page for Wallets in Fusion Middleware Control."](#)
2. Click **Import**.
3. In the Import Certificate dialog, you can select either a certificate or a trusted certificate.
4. There are two ways to do the import:
 - Paste the Base64-encoded certificate or trusted certificate in the text box.
 - Use the file selector to browse your file system to locate a file containing the Base64-encoded certificate or trusted certificate.



5. Click **OK**.

7.4.7.6 Importing a Certificate or a Trusted Certificate Using WLST

Note: You cannot use the WLST command-line tool to import DER-encoded certificates or trusted certificates into an Oracle wallet. Use `orapki` command-line tool instead.

Assuming the instance name is `inst1`, use this command to import a certificate into a wallet:

```
importWalletObject('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'Certificate',#'/tmp/cert.txt')
```

where `password` is the password for this wallet and `/tmp/cert.txt` is the file that contains BASE64 encoded certificate.

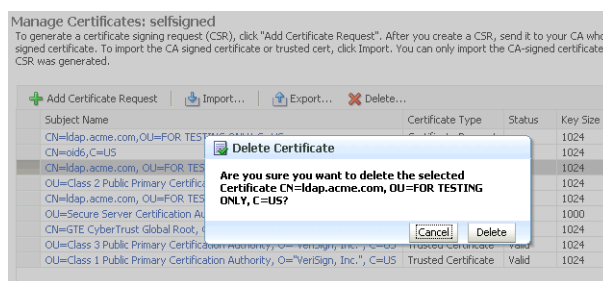
To import a trusted certificate, replace `Certificate` in the above command with `TrustedCertificate`.

See Also: [Section 6.9.12](#)

7.4.7.7 Deleting a Certificate Request, a Certificate, or a Trusted Certificate Using Fusion Middleware Control

Take these steps to delete a CR, a certificate, or a trusted certificate:

1. Navigate to the Certificate Management page. See [Section 7.4.6, "Accessing the Certificate Management Page for Wallets in Fusion Middleware Control."](#)
2. Select the row containing the certificate request, certificate or trusted certificate.
3. Click **Delete**.
4. A dialog box appears, requesting confirmation.



5. Click **Yes**.
6. The object no longer appears in the Manage Certificates list.

7.4.7.8 Deleting a Certificate Request, a Certificate, or a Trusted Certificate Using WLST

Assuming the instance name is `inst1`, use this command to delete a certificate:

```
removeWalletObject('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'Certificate',#'subject_dn')
```

where `password` is the password for this wallet and `subject_dn` is the distinguished name of the certificate being deleted.

To delete a certificate request or trusted certificate, replace `Certificate` in the above command with `CertificateRequest` or `TrustedCertificate`.

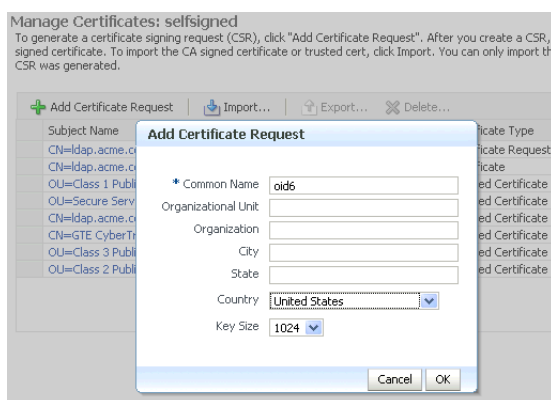
See Also: [Section 6.9](#)

7.4.7.9 Converting a Self-Signed Certificate into a Third-Party Certificate Using Fusion Middleware Control

A self-signed certificate residing in a wallet can be converted into a third-party certificate signed by a certificate authority (CA). Take these steps to perform the task:

Note: The steps are illustrated for use with Oracle Internet Directory, and similar steps are applicable for generating wallets to use with Oracle HTTP Server.

1. From the navigation pane, locate your component instance.
2. Navigate to `component_name`, the **Security**, then **Wallets**.
3. From the list of wallets, select the wallet that contains the self-signed certificate.
4. The Manage Certificates page appears. It contains the list of certificates in the wallet.
5. A new certificate request must be generated for the self-signed certificate that is to be converted. Select the self-signed certificate and click **Add Certificate Request**. A dialog box appears:



Note: The common name entered here should match the host name of the server to which clients will connect; this helps to prevent problems of the type mentioned in [Section 7.4.8.2](#).

6. Enter the certificate request (CR) details and click **OK**.

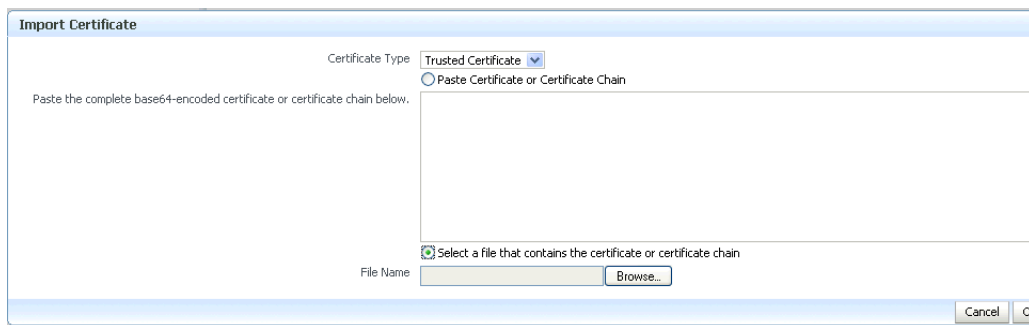
The CR is generated. You can either:

- Copy and paste the Base64-encoded certificate request to a file.
 - Export it directly to a file with the **Export Certificate Request** button.
7. Submit the certificate request file to a certificate authority to generate a certificate. This is an offline procedure that you can execute in accordance with your local policy for obtaining certificates.
 8. The CA signs the certificate request and generates a certificate. The CA will return you one of the following:
 - A single file containing both the newly generated certificate and its own CA certificate in `pkcs7` format
 - Multiple files, one of which will be the newly generated certificate and the other(s) its own CA certificate (or certificates, if there is a chain)
 9. Use **Import** to import these files into your wallet:
 - If you received a single file from the CA, import it as a trusted certificate, using an alias that matches the alias of the self-signed certificate you are replacing (from Step 3).
 - If you received two or more files:
 - Import the file(s) containing the CA certificate or certificate chain as trusted certificate(s) (use an alias that is unique in the wallet).
 - Import the certificate file as a certificate (using an alias that matches the alias of the self-signed certificate you are replacing).

Note: The order is important: you must import the trusted certificate first, followed by the certificate.

For more information on how to identify the correct Trusted CA Certificate or the correct chain of Trusted CA certificates, see Doc ID 1368940.1 in the Oracle Technology Network Knowledge Base.

Here the CA returned a single file, which is imported as a trusted certificate:



10. After import, the certificate issued by the CA replaces the self-signed certificate.

7.4.7.10 Converting a Self-Signed Certificate into a Third-Party Certificate Using WLST

See Also: [Section 6.9](#)

Follow these steps to convert a self signed certificate to a third-party certificate using WLST:

1. Add a certificate request, for example:

```
addCertificateRequest('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'subject_dn',#'key_size')
```

2. Export the certificate request:

```
exportWalletObject('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'CertificateRequest',#'/tmp',#'subject_dn')
```

3. Submit the certificate request `/tmp/base64.txt` to a certificate authority. The CA will return a newly generated certificate and its own certificate, either as one file in PKCS#7 format or as multiple files, one of which will be the newly generated certificate and the other(s) its own CA certificate (or certificates, if there is a chain).

4. If you receive a single file from the CA, run the following command

```
importWalletObject('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'TrustedChain',#'/tmp/cert.txt')
```

where `password` is the password for this wallet and `/tmp/cert.txt` is the file that the CA returned and contains BASE64 encoded PKCS#7.

If you receive two or more files from the CA, import the file(s) containing the CA certificate or certificate chain as trusted certificate(s) first, followed by the newly generated certificate. For example when two files are returned:

```
importWalletObject('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'TrustedCertificate',#'/tmp/cacert.txt')
#
importWalletObject('inst1',#'ohs1',#'ohs',#'selfsigned',#'password',#
'Certificate',#'/tmp/cert.txt')
```

where `password` is the password for this wallet, `/tmp/cert.txt` is the file that the CA returned and contains BASE64 encoded certificate and `/tmp/cacert.txt` is the file containing the BASE64 encoded CA certificate.

For more information on how to identify the correct Trusted CA Certificate or the correct chain of Trusted CA certificates, see Doc ID 1368940.1 in the Oracle Technology Network Knowledge Base.

7.4.8 Wallet and Certificate Maintenance

This section contains the following administration topics:

- [Location of Wallets](#)
- [Effect of Host Name Change on a Wallet](#)
- [Changing a Self-Signed Wallet to a Third-Party Wallet](#)
- [Replacing an Expiring Certificate in a Wallet](#)

7.4.8.1 Location of Wallets

This section describes the location of wallets and provides maintenance details.

Root Directory for an Oracle HTTP Server Wallet

The root directory for Oracle HTTP Server wallets is:

```
$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance_name/keystores
```

This root directory contains subdirectories with wallet names, and these subdirectories contain the actual wallet files.

For example, assuming `ohs_instance1` contains two wallets named `ohs1` and `ohs2`, respectively. `ohs1` is a password-protected wallet, and `ohs2` is an auto-login-only wallet. A sample structure could look like this:

```
$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance1#  
/keystores/ohs1/cwallet.sso
```

```
$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance1#  
/keystores/ohs1/ewallet.p12
```

```
$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance1#  
/keystores/ohs2/cwallet.sso
```

7.4.8.2 Effect of Host Name Change on a Wallet

Typically, the wallet DN is based on the host name of the server where the wallet is used.

For example, if a wallet is being created for the Oracle HTTP Server `my.example.com`, then the DN of the certificate in this Oracle HTTP Server wallet will be something like `"CN=my.example.com,O=organization#name"`.

This synchronization is required because most clients do host name verification during the SSL handshake.

Clients that perform host name verification include Web browsers and Oracle HTTPClient, among others. If the host name of the server does not match that of the certificate DN, then:

- A clear warning will be displayed (in the case of browser clients).
- There may be SSL handshake failure (in the case of other clients).

Thus, when you have a wallet on a server that is accepting requests from clients, you must ensure that whenever the host name of this server changes, you also update the certificate in the wallet.

You can do this by requesting a new certificate with a new DN (based on the new host name).

For a Production Wallet

The steps are:

- Generate a new request with the new DN (based on new host name).
- Send this request to a certificate authority (CA).
- Get back a new certificate from the CA.
- Delete the older certificate and certificate request from the wallet.
- Import the new certificate.

See [Section 7.4.4](#) for details about these operations.

For a Self-signed Wallet

The steps are:

- Delete the existing wallet.

- Create a new wallet with a self-signed certificate using the new DN (based on the new host name).

See [Section 7.4.4](#) for details about these operations.

For both production and self-signed wallets, once the new certificate is available in the wallet, you need to ensure that it is imported into all the component wallets where it needs to be trusted. For example, if Oracle WebLogic Server is SSL-enabled and the certificate for Oracle WebLogic Server changed due to a host name change, then you need to import its new certificate into the Oracle HTTP Server wallet so that it can trust its new peer.

7.4.8.3 Changing a Self-Signed Wallet to a Third-Party Wallet

You can convert a self-signed wallet into a third-party wallet, one that contains certificates signed by a trusted Certificate Authority (CA).

Assuming a self-signed wallet named `MYWallet`, containing a certificate with DN as `"CN=my.example.com,O=example"`, take these steps to convert it into a third-party wallet:

1. Remove the user certificate `"CN=my.example.com,O=example"` from the wallet.
2. Remove the trusted certificate `"CN=my.example.com,O=example"` from the wallet (this has the same DN as the user certificate, but is a separate entity nonetheless).
3. Export the certificate request `"CN=my.example.com,O=example"` from the wallet and save it to a file.
4. Give this certificate request file to a third-party certificate authority (CA) such as Verisign.
5. The CA will return one of the following:
 - A user certificate file and its own certificate file
 - A single file with a certificate chain consisting of a user certificate and its own certificate
6. Import the above file(s) into the wallet.

See [Section 7.4.4](#) for details about these operations.

7.4.8.4 Replacing an Expiring Certificate in a Wallet

An expiring certificate should be replaced before it actually expires to avoid or reduce application downtime.

The steps for replacing an expiring certificate are as follows:

1. Export the certificate request from the wallet (this is the same request for which the current expiring certificate was issued).
2. Provide this certificate request to the third-party Certificate Authority (CA) for certificate issuance. The validity date of the new certificate should be earlier than the expiration date of the current certificate. This overlap is recommended to reduce downtime.

Note: Steps 1 and 2 are not required when the third-party CA already maintains the certificate request in a repository. In that case, simply request the CA to issue a new certificate for that certificate request.

3. Remove the existing certificate (the one that is about to expire) from the wallet.
4. Import the newly issued certificate into the wallet.
To reduce downtime, remove the previous certificate and import the new certificate in the overlap period when the new certificate has become valid and the older one has not yet expired.
5. If the new certificate was issued by a CA other than the one that issued the original certificate, you may also need to import the new CA's trusted certificate before importing the newly issued certificate.

See [Section 7.4.4](#) for details about these operations.

Part IV

Deploying Applications

This part describes the deployment process and how to deploy applications to Oracle Fusion Middleware.

Part IV contains the following chapters:

- [Chapter 8, "Understanding the Deployment Process"](#)
- [Chapter 9, "Deploying Applications"](#)

Understanding the Deployment Process

Before you deploy Oracle Fusion Middleware applications, such as Java EE applications, you should understand the deployment process, such as designing and developing applications and deploying those applications to Managed Servers.

This chapter describes the following sections:

- [What Is a Deployer?](#)
- [General Procedures for Moving from Application Design to Production Deployment](#)
- [Diagnosing Typical Problems](#)

8.1 What Is a Deployer?

A user in the role of **deployer** is responsible for deploying applications, such as Java EE applications, and ADF applications, to WebLogic Server instances or clusters.

A user who is functioning as a deployer should be granted the Oracle WebLogic Server deployer security role. The deployer security role allows deployment operations, as well as viewing the server configuration and changing startup and shutdown classes. To grant this role to a user, use the Oracle WebLogic Server Administration Console. See "Managing Security Roles" in the Oracle WebLogic Server Administration Console Help for more information.

8.2 General Procedures for Moving from Application Design to Production Deployment

This section describes the general procedures involved in moving from application design and development to deployment in a production environment. It contains the following topics:

- [Designing and Developing an Application](#)
- [Deploying an Application to Managed Servers](#)
- [Automating the Migration of an Application to Other Environments](#)

8.2.1 Designing and Developing an Application

In many cases, developers use Oracle JDeveloper to create their applications. Oracle JDeveloper is an integrated development environment (IDE) for building service-oriented applications using the latest industry standards for Java, XML, Web services, portlets, and SQL. JDeveloper supports the complete software development

life cycle, with integrated features for modeling, coding, debugging, testing, profiling, tuning, and deploying applications.

In this environment, you use the integrated Oracle WebLogic Server, which is packaged with Oracle JDeveloper for testing your applications.

For information about developing your applications, see:

- *Developing Applications for Oracle WebLogic Server*
- *Developing Fusion Web Applications with Oracle Application Development Framework*

8.2.2 Deploying an Application to Managed Servers

After you have designed and tested your application with the integrated Oracle WebLogic Server, you can deploy the application to a Managed Server instance. For example, you may have installed Oracle WebLogic Server and configured a domain, including a Managed Server, in your production environment and you want to deploy the application to that Managed Server.

The following books provide specific information about deploying the different types of applications:

- For Java EE applications, see *Deploying Applications to Oracle WebLogic Server*
- For Oracle ADF, see *Administering Oracle ADF Applications*

This section provides an outline of the major steps involved when you migrate your application from the integrated Oracle WebLogic Server to an environment separate from the development environment. Those general steps are:

1. Package the application:
 - For Java EE applications, you package the application in an EAR file. See "Preparing Applications and Modules for Deployment" in *Deploying Applications to Oracle WebLogic Server*
 - For Oracle ADF, you package the application in an EAR file. See "What You May Need to Know About EAR Files and Packaging" in *Developing Fusion Web Applications with Oracle Application Development Framework*
2. Set up your environment. This includes:
 - Installing and configuring a domain and a Managed Server that is configured with the correct domain template. The appropriate domain template is applied when you create the domain using the Configuration Wizard.

For more information about installing and configuring for specific components, see:

 - For Oracle ADF: "How to Install the ADF Runtime to the WebLogic Installation" in *Administering Oracle ADF Applications*
 - Creating any necessary schemas in an existing database. See *Creating Schemas with the Repository Creation Utility*.
 - Registering the MDS Repository with the Oracle WebLogic Server domain, if your application uses the MDS Repository. For example, some ADF applications involve customizations using MDS. See [Section 13.3.2.1.1](#) for information about registering the MDS Repository.
3. If your application uses a database, set up the JDBC data sources.

For more information about setting up the JDBC data sources, see:

- For pure Java EE applications: *Administering JDBC Data Sources for Oracle WebLogic Server*
 - For Oracle ADF: "How to Create a JDBC Data Source for Oracle WebLogic Server" in *Administering Oracle ADF Applications*
4. Create a connection to the target Managed Server.
- From Oracle JDeveloper, you can deploy your applications to Managed Server instances that reside outside JDeveloper. To do this, you must first create a connection to the server instance to which you want to deploy your application.
- For more information, see:
- For Oracle ADF: "How to Create a Connection to the Target Application Server" in *Developing Fusion Web Applications with Oracle Application Development Framework*
5. Create a configuration plan or deployment plan, which contains information about environment-specific values, such as JDBC connection strings or host names of various servers. For more information, see:
- For pure Java EE applications: "Creating a New Deployment Plan to Configure an Application" in *Deploying Applications to Oracle WebLogic Server*
6. Migrate application security, such as credentials, identities, and policies. For more information, see:
- For pure Java EE applications: "Migrating Security Data" in *Administering Security for Oracle WebLogic Server*
 - For Oracle ADF: "Preparing the Secure Application for Deployment" in *Developing Fusion Web Applications with Oracle Application Development Framework*
7. Create a deployment profile. A **deployment profile** packages or archives a custom ADF and associated files so that the application can be deployed to an Oracle WebLogic Server Managed Server instance. Deployment profiles are created at the project and application level.
- For more information, see:
- For Oracle ADF: "How to Create Deployment Profiles" in *Developing Fusion Web Applications with Oracle Application Development Framework*
8. Deploy the application to a Managed Server.
- For more information, see:
- For pure Java EE applications: "Exporting an Application for Deployment to New Environments" in *Deploying Applications to Oracle WebLogic Server*
 - For Oracle ADF: "Deploying the Application" in *Developing Fusion Web Applications with Oracle Application Development Framework*

8.2.3 Automating the Migration of an Application to Other Environments

You can automate the migration of an application by using WLST or ant scripts. This makes it easier to deploy your application to multiple environments or Managed Servers and to deploy updated versions of the application.

For more information about using scripts to migrate an application to other environments, see:

- For pure Java EE applications: "Using the WebLogic Scripting Tool" in *Understanding the WebLogic Scripting Tool*
- For Oracle ADF: "Deploying Using Scripts and Ant" in *Administering Oracle ADF Applications*

8.3 Diagnosing Typical Problems

The following describes some of the typical problems that you may encounter when you deploy an application to a Managed Server:

- Connection information. Ensure that you have correctly configured the connection to the target Managed Server. See Step 4 in [Section 8.2.2](#).
- Data sources. Ensure that you have correctly configured JDBC data sources. See Step 3 in [Section 8.2.2](#).
- Security configuration. Ensure that you have migrated application security, such as credentials, identities, and policies. See Step 6 in [Section 8.2.2](#).

Deploying Applications

Deployment is the process of packaging application files as an archive file and transferring them to a target application server. This chapter describes how to deploy applications, to Oracle Fusion Middleware.

It contains the following sections:

- [Overview of Deploying Applications](#)
- [Understanding and Managing Data Sources](#)
- [Deploying, Undeploying, and Redeploying Java EE Applications](#)
- [Deploying, Undeploying, and Redeploying Oracle ADF Applications](#)
- [Managing Deployment Plans](#)
- [About the Common Deployment Tasks in Fusion Middleware Control](#)
- [Changing MDS Configuration Attributes for Deployed Applications](#)

9.1 Overview of Deploying Applications

Oracle WebLogic Server provides a Java EE-compliant infrastructure for deploying, undeploying, and redeploying Java EE-compliant applications and modules.

The following topics describe:

- [What Types of Applications Can You Deploy?](#)
- [Understanding Deployment, Redeployment, and Undeployment](#)

9.1.1 What Types of Applications Can You Deploy?

You can deploy the following into Oracle WebLogic Server:

- A complete Java EE application packaged as an Enterprise Archive (EAR) file.
- Standalone modules packaged as Java Archive files (JARs) containing Web services, Enterprise JavaBeans (EJBs), application clients (CARs), or resource adapters (RARs).
- An ADF application. Oracle Application Development Framework (Oracle ADF) is an end-to-end application framework that builds on Java Platform, Enterprise Edition (Java EE) standards, and open-source technologies to simplify and accelerate implementing service-oriented applications.

A Metadata Archive (MAR) is a compressed archive of selected metadata, such as the application-level deployment profile, for an application. A MAR is used to deploy

metadata content to the metadata service (MDS) repository. ADF applications use a MAR as a container for content that is deployed to the MDS Repository.

Note: If your application uses password indirection in the application-level data source, you cannot use Fusion Middleware Control to deploy the application. The section "Deploying an Application to an EAR File to run on Oracle WebLogic Server" in the Oracle JDeveloper Help describes how to change the settings of the application to be able to deploy the application using Fusion Middleware Control.

You can use Fusion Middleware Control, Oracle WebLogic Server Administration Console, Oracle JDeveloper, or the command line to deploy, undeploy, or redeploy an application. Which method you use depends on the type of application, as described in [Table 9-1](#).

Table 9-1 Tools to Deploy Applications

Type of Application	Tools to Use
Pure Java EE application	Oracle WebLogic Server Administration Console Fusion Middleware Control: Deployment Wizard Oracle JDeveloper WLST command line
ADF application	Fusion Middleware Control: Deployment Wizard Oracle JDeveloper WLST command line

If your application uses an MDS Repository, you must register the repository with the Oracle WebLogic Server domain before you deploy your application. Applications such as custom Java EE applications developed by your organization and some Oracle Fusion Middleware component applications, such as Oracle Web Services Manager, use an MDS Repository. For information about the MDS Repository and registering the repository, see [Section 13.3](#).

Note: If your application contains an application-level credential store, and you are moving the application from a test to a production environment, you must reassociate the credential store, as described in "Reassociating the Domain Policy Store" in the *Securing Applications with Oracle Platform Security Services*.

9.1.2 Understanding Deployment, Redeployment, and Undeployment

When you deploy an application, you deploy it to the application server for the first time.

When you redeploy an application, you can:

- Redeploy a new version of the application; the previous version is still available, but the state is set to "Retired."

This is known as the production redeployment strategy. Oracle WebLogic Server automatically manages client connections so that only new client requests are directed to the new version. Clients already connected to the application during

the redeployment continue to use the older version of the application until they complete their work, at which point Oracle WebLogic Server automatically retires the older application.

- Redeploy the same version of the application or redeploy an application that is not assigned a version; the application version you select is replaced with the new deployment.
- Redeploy a previous version of the application; the earlier, retired version is set to "Active" and the later version is set to "Retired."

When you undeploy an application, Oracle WebLogic Server stops the application and removes staged files from target servers. It does not remove the original source files used for deployment.

9.2 Understanding and Managing Data Sources

The following topics describe data sources and how to manage them:

- [Understanding Data Sources](#)
- [Creating and Managing JDBC Data Sources](#)

9.2.1 Understanding Data Sources

A **data source** is a Java object that application components use to obtain connections to a relational database. Specific connection information, such as the URL or user name and password, are set on a data source object as properties and do not need to be explicitly defined in an application's code. This abstraction allows applications to be built in a portable manner, because the application is not tied to a specific back-end database. The database can change without affecting the application code.

Applications use the Java Naming and Directory Interface (JNDI) API to access a data source object. The application uses a JNDI name that is bound to the data source object. The JNDI name is logical and can be mapped to any data source object. Like data source properties, using JNDI provides a level of abstraction, since the underlying data source object can change without any changes required in the application code. The end result is the details of accessing a database are transparent to the application.

See *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about data sources.

When you configure certain Oracle Fusion Middleware components using the Oracle WebLogic Server Configuration Wizard, you specify the data source connection information. If the components use the MDS Repository, the Configuration Wizard prepends "mds-" to the data source name to indicate that the data source is a system data source used by MDS Repository.

See *Creating Domains Using the Configuration Wizard* for information about specifying data sources with the Configuration Wizard.

If you are using Oracle Real Application Clusters (Oracle RAC) or Oracle Fusion Middleware Cold Failover Cluster, you must configure one of the following types of data sources:

- Multi data sources

To use multi data sources, you must use the Oracle WebLogic Server Administration Console. Note that if you create a multi data source and you add an existing MDS data source to it, the data source you added is no longer considered a valid MDS Repository. The repository is not displayed in Fusion

Middleware Control or Oracle WebLogic Server Administration Console. For example, the MDS Repository is not listed in the Fusion Middleware Control navigation pane and is not displayed as a choice for a target metadata repository when you deploy an application.

- GridLink data sources

To use GridLink data sources, you can use the Oracle WebLogic Server Administration Console or Fusion Middleware Control, as described in [Section 9.2.2.5](#).

See *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about configuring multi data sources and GridLink data sources

9.2.2 Creating and Managing JDBC Data Sources

You can create and manage JDBC data sources using the following management tools:

- The Oracle WebLogic Server Administration Console
- The WebLogic Scripting Tool (WLST)
- Fusion Middleware Control

To create an MDS data source manually, you should use Fusion Middleware Control or WLST to set the correct attributes for the data source. The MDS data source is displayed in the navigation pane in Fusion Middleware Control and in the domain structure in the Administration Console. If your application uses an MDS Repository, you must register the repository with the Oracle WebLogic Server domain before you deploy your application. For information about the MDS Repository and registering the repository, see [Section 13.3](#).

Note: When you create the data source, you must use the MDS schema created by the Repository Creation Utility (RCU), not other schemas.

Although it is not recommended, you can also use the Oracle WebLogic Server Administration Console to create a MDS data source. If you do, note the following:

- You must prefix the data source name with "mds-" if you intend it to be used with MDS Repository.
- You must target the data source to the Administration Server and to all Managed Servers to which you are deploying applications that need the data source.
- You must turn off global transactions.

See *Administering JDBC Data Sources for Oracle WebLogic Server* for information about creating and managing a data source using the Oracle WebLogic Server Administration Console or WLST and for more information about configuring multiple data sources.

The following topics describe how to create and manage JDBC data sources with Fusion Middleware Control:

- [Creating a JDBC Data Source Using Fusion Middleware Control](#)
- [Editing a JDBC Data Source Using Fusion Middleware Control](#)
- [Monitoring a JDBC Data Source Using Fusion Middleware Control](#)
- [Controlling a JDBC Data Source Using Fusion Middleware Control](#)

- [Creating a GridLink Data Source Using Fusion Middleware Control](#)

9.2.2.1 Creating a JDBC Data Source Using Fusion Middleware Control

To create a JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

The JDBC Data Sources page is displayed, as shown in the following figure:

Name	JNDI Name	Type	Targets
LocalSvcTblDataSource	jdbc/LocalSvcTblDataSource	Generic	AdminServer
mds-owsm	jdbc/mds/owsm	Generic	AdminServer
opss-audit-DBDS	jdbc/AuditAppendDataSource	Generic	AdminServer
opss-audit-viewDS	jdbc/AuditViewDataSource	Generic	AdminServer
opss-data-source	jdbc/OpssDataSource	Generic	AdminServer

2. From **Create**, select **Generic Data Source**.
3. Follow the instructions in the wizard to set the properties of the data source and to target the data source for one or more of the Managed Servers in the domain.

For help on individual fields and properties, use your mouse to give focus to a field. Fusion Middleware Control displays a popup definition of the field.

Note that the data source properties you define in Fusion Middleware Control are similar to those you define when creating data sources in the Oracle WebLogic Server Administration Console. As a result, you can also refer to "Creating a JDBC Data Source" in *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about the data source properties.

9.2.2.2 Editing a JDBC Data Source Using Fusion Middleware Control

To edit an existing JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

The JDBC Data Sources page is displayed.

2. Click the data source that you want to edit.

The page for that particular JDBC Data Source is displayed.

3. Use the tabs on this page to modify the properties of the selected data source.

For help on individual fields and properties, use your mouse to give focus to a field. Fusion Middleware Control displays a popup definition of the field.

Note that the data source properties you edit in Fusion Middleware Control are similar to those you edit when editing data sources in the Oracle WebLogic Server Administration Console. As a result, you can also refer to "Creating a JDBC Data Source" in *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about the data source properties.

9.2.2.3 Monitoring a JDBC Data Source Using Fusion Middleware Control

To monitor a JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

The JDBC Data Sources page is displayed.

2. Select the data source that you want to monitor.
3. Click **Monitoring** to display the Monitor JDBC Data Source page.

This page shows the current instances of the selected data source.

Note that only data sources that are targeted to a running Managed Server are shown on this page. If a specific data source is not listed on the monitoring page, then edit the data source to be sure it is targeted to a running Managed Server.

4. For each data source instance, review the performance metrics.

9.2.2.4 Controlling a JDBC Data Source Using Fusion Middleware Control

To start, stop, suspend, resume, or clear the statement cache for a JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

The JDBC Data Sources page is displayed.

2. Select the data source that you want to edit.
3. Select the Control tab.

Note that only data sources that are targeted to a running Managed Server are shown on this page. If a specific data source is not listed on the control page, edit the data source to be sure that it is targeted to a running Managed Server.

4. Select the instance and click Start, Stop, Resume, Suspend, Shrink, Reset, Clear Statement Cache to control or change the state of the selected JDBC data source.

Note that the commands you select on this page are similar to those available when you are managing data sources in the Oracle WebLogic Server Administration Console. Refer to "Managing WebLogic JDBC Resources" in *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about the JDBC data source control options.

9.2.2.5 Creating a GridLink Data Source Using Fusion Middleware Control

A single GridLink data source provides connectivity between WebLogic Server and an Oracle Database service targeted to an Oracle RAC cluster. For detailed information about GridLink data sources, see "Creating a GridLink Data Source" in *Administering JDBC Data Sources for Oracle WebLogic Server*.

To create a Grid Link data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

The JDBC Data Sources page is displayed.

2. From **Create**, select **GridLink Data Source**.
3. Follow the instructions in the wizard to set the properties of the data source and to target the data source for one or more of the Managed Servers in the domain.

For help on individual fields and properties, use your mouse to give focus to a field. Fusion Middleware Control displays a popup definition of the field.

Note that the data source properties you define in Fusion Middleware Control are similar to those you define when creating data sources in the Oracle WebLogic Server Administration Console. As a result, you can also refer to "Creating a

GridLink Data Source" in *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about the data source properties.

9.3 Deploying, Undeploying, and Redeploying Java EE Applications

You can use Fusion Middleware Control, Oracle WebLogic Server Administration Console, Oracle JDeveloper, or the command line to deploy, undeploy, or redeploy a Java EE application. The following topics describe using Fusion Middleware Control and the command line to accomplish these tasks:

- [Deploying Java EE Applications](#)
- [Undeploying Java EE Applications](#)
- [Redeploying Java EE Applications](#)

See *Deploying Applications to Oracle WebLogic Server* for information about deploying using Oracle WebLogic Server Administration Console and the WLST command line.

9.3.1 Deploying Java EE Applications

You can deploy an application to a WebLogic Server Managed Server instance or a cluster. This section describes how to deploy an application to a Managed Server.

9.3.1.1 Deploying Java EE Applications Using Fusion Middleware Control

To deploy a Java EE application to a Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the server in which you want to deploy the application.
The server home page is displayed.
3. From the WebLogic Server menu, choose **Control**, then **Deployments**.
The Deployments page is displayed.
4. Click **Deploy** to open the Deploy Java EE Application Assistant.
The Select Archive page is displayed, as shown in the following figure:

Deploy Java EE Application : Select Archive Back Step 1 of 4 Next Cancel

Specify the application or the exploded directory. Optionally you can specify a deployment plan.

Archive or Exploded Directory
 Java EE archives, Web Modules (WAR files), EJB Modules (EJB JAR files), Resource Adapter Modules (RAR files), Coherence Archives (GAR files), JDBC Modules, JMS Modules, and library files (Jar files) can be deployed. You can also deploy an exploded archive that is present on the server where Enterprise Manager is running.

Archive is on the machine where this Web browser is running.
 Archive or exploded directory is on the server where Enterprise Manager is running.

Deployment Plan
 The deployment plan is a file that contains the deployment settings for an application. You can use a previously saved deployment plan for this application. Later in the deployment process, you can optionally edit the deployment plan and save it for a future deployment of this application. If you do not have a deployment plan, one will be created automatically during the deployment process when deployment configuration is done. The deployment plan is not applicable when you deploy a library.

Create a new deployment plan when deployment configuration is done.
 Deployment plan is on the machine where this Web browser is running.
 Deployment plan is on the server where Enterprise Manager is running.

Deployment Type
 The archive or exploded directory can be deployed as a regular application or a library. Application libraries are deployments that are available for other deployments to share. Libraries should be available on all of the targets running their referencing applications. The deployment type option will be set as library automatically when you deploy a library file (Jar file).

Deploy this archive or exploded directory as an application
 Deploy this archive or exploded directory as a library

Information
 Use this page to deploy Java EE applications that require Oracle Metadata Services (MDS) or that take advantage of the Oracle Application Development Framework (Oracle ADF).
 If your application is a SOA composite use the SOA Composite deployer wizard.
 If your application is not a SOA composite or it does not require an repository or ADF connections, the you can deploy your application using this wizard or the Oracle WebLogic Server Administration Console.

5. In the Archive or Exploded Directory section, you can select one of the following:
 - **Archive is on the machine where this browser is running.** Enter the location of the archive or click **Browse** to find the archive file.
 - **Archive or exploded directory is on the server where Enterprise Manager is running.** Enter the location of the archive or click **Browse** to find the archive file.
6. In the Deployment Plan section, you can select one of the following:
 - **Create a new deployment plan when deployment configuration is done.**
 - **Deployment plan is on the machine where this Web browser is running.** If you select this option, enter the path to the plan.
 - **Deployment plan is on the server where Enterprise Manager is running.** If you select this option, enter the path to the plan.
7. In the Deployment Type section, you can select one of the following:
 - **Deploy this archive or exploded directory as an application**
 - **Deploy this archive or exploded directory as a library**
8. Click **Next**.

The Select Target page is displayed.

9. Select the target to which you want to deploy the application. The Administration Server, Managed Servers, and clusters are listed. You can select a cluster, one or more Managed Servers in the cluster, or a Managed Server that is not in a cluster. Although the Administration Server is shown in the list of targets, you should not deploy an application to it. The Administration Server is intended only for

administrative applications such as the Oracle WebLogic Server Administration Console.

10. Click Next.

The Application Attributes page is displayed.

11. In the Application Attributes section, for **Application Name, enter the application name.**

12. In the Context Root of Web Modules section, if the Web module does not have the context root configured in the application.xml file, you can specify the context root for your application. The **context root is the URI for the Web module. Each Web module or EJB module that contains Web services may have a context root.**

13. In the Distribution section, you can select one of the following:

- **Install and start application (servicing all requests)**
- **Install and start application in administration mode (servicing only admin requests)**
- **Install only. Do not start.**

14. You can expand Other Options, which provides the following for Application Source Accessibility:

- Use the defaults defined by the deployment's targets. Recommended selection.
- Copy this application onto every target. During deployment, the files will be copied automatically to the Managed Servers to which the application is targeted.

15. In Other Options, you can also select one of the following for Deployment Plan Source Accessibility:

- Use the same accessibility as the application.
- Copy the deployment plan onto every target. During deployment, the files will be copied automatically to the Managed Servers to which the application is targeted.
- Make the deployment plan accessible from the source location that it will be deployed on. You must ensure that each target can reach the location.

16. Click Next.

The Deployment Wizard, Deployment Settings page is displayed.

17. On this page, you can perform common tasks before deploying your application or you can edit the deployment plan or save it to a disk.

See [Section 9.6](#) for more detailed information about these tasks.

Depending on the type of application, in the Deployment Tasks section, you can:

- **Configure Web modules:** Click **Go to Task** in the Configure Web Modules row. The Configure Web Modules page is displayed. Click **Configure General Properties** to view and edit the general configuration for the Web Module or **Map Resource References** to map the resource references.

For example, you can change the session invalidation interval or the maximum age of session cookies.

- **Configure EJB modules:** Click **Go to Task** in the Configure EJB modules row to set standard EJB deployment descriptor properties. The Configure EJB Modules page is displayed. Click **Configure EJB Properties** to view and edit

the general configuration for the EJBs or **Map Resource References** to map the resource preferences.

For example, you can configure the maximum number of beans in the free pool or the network access point.

- Configure application security: Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed, as described in [Section 9.6](#).
- Configure persistence: Click **Go to Task** in the Configure Persistence row to configure Java Persistent API (JPA) persistence units.

18. Expand **Deployment Plan**.

You can edit and save the deployment plan, if you choose. If you edit the deployment plan and change descriptor values, those changes are saved to the deployment plan. In addition, the following configurations are saved to the deployment plan:

- Application attributes
- Web module configuration
- EJB configuration

Application attributes related to MDS are stored in the file `adf-config.xml`. Application security attributes are stored in `weblogic-application.xml`.

Fusion Middleware Control updates the relevant files and repackages the `.ear` file.

19. Click **Deploy**.

Fusion Middleware Control displays processing messages.

20. When the deployment is completed, click **Close**.

To deploy an application to multiple servers at the same time, navigate to the domain. Then, from the WebLogic Domain menu, select **Application Deployment**, then **Deploy**. The deployment wizard displays a page where you can select the servers.

To deploy an application to a cluster, select the cluster. Then, from the Cluster menu, select **Application Deployment**, then **Deploy**.

9.3.1.2 Deploying Java EE Applications Using WLST

You can deploy an application using the WLST command line. To deploy a Java EE application when WLST is connected to the Administration Server, you use the WLST command `deploy`, using the following format:

```
deploy(app_name, #path# [, targets] # [, stageMode] # [, planPath] # [, options])
```

You must invoke the `deploy` command on the computer that hosts the Administration Server.

For example, to deploy the application `mainWebApp`:

```
deploy("myApp", "/scratch/applications/wlserver/samples/server/examples/build/mainWebApp")
```

You can also deploy the application using the `weblogic.deployer`, as shown in the following example:

```
java#weblogic.Deployer#-adminurl#http://localhost:7001#
###-user#username#-password#password#-deploy#
###-name#myApp#c:\localfiles\mainWebApp
```

```
###-plan#c:\localfiles\productionEnvPlan.xml
```

See Also:

- "Deployment Tools" in *Deploying Applications to Oracle WebLogic Server* for more information about using WLST to deploy applications
- *WLST Command Reference for WebLogic Server*

9.3.2 Undeploying Java EE Applications

You can undeploy an application or a specific version of an application from a WebLogic Server Managed Server instance or a cluster. This section describes how to undeploy an application from a Managed Server. If an application has been deployed to multiple servers, when you undeploy it using Fusion Middleware Control, the application is undeployed from all the servers.

9.3.2.1 Undeploying Java EE Applications Using Fusion Middleware Control

To undeploy a Java EE application from a Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments**.
2. Select the application to undeploy.
The application home page is displayed.
3. From the Domain Application Deployment menu, choose **Deployments**.
4. Select the application.
5. In Confirmation page, click **Undeploy**.
Processing messages are displayed.
6. When the operation completes, click **Close**.

Alternatively, you can navigate to the domain, Managed Server, or cluster. Then, from the target's menu, choose **Application Deployment**, then **Undeploy**. In the Select Application page, select the application you want to undeploy.

9.3.2.2 Undeploying Java EE Applications Using WLST

You can undeploy an application using the WLST command line. To undeploy a Java EE application when WLST is connected to the Administration Server, you use the WLST command `undeploy`, using the following format:

```
undeploy(app_name, #path#[, targets]#[, options])
```

You must invoke the undeploy command on the computer that hosts the Administration Server.

For example, to undeploy the application `businessApp` from all target servers and specify that WLST wait 60,000 ms for the process to complete:

```
wls:/mydomain/serverConfig>undeploy('businessApp', #timeout=60000)
```

9.3.3 Redeploying Java EE Applications

You can redeploy a new version of an updated application, redeploy the same version, or redeploy a non-versioned application. You can redeploy an application to a cluster or a Managed Server.

The following sections describe how to redeploy an application to a Managed Server:

- [Redeploying Java EE Applications Using Fusion Middleware Control](#)
- [Redeploying Java EE Applications Using WLST](#)

If you are redeploying a non-versioned application or a versioned application with the same version, note the following:

- The file name and path for the archive you are redeploying must be identical to the file name and path you used when you initially deployed the application.
For example, if the file name and path of the original application was `/dua0/staging/myApp.ear`, then the revised application must be `/dua0/staging/myApp.ear`.
- If you initially deployed the application using the Oracle WebLogic Server Administration Console or WLST or other management tools other than Fusion Middleware Control, then you cannot redeploy the application using Fusion Middleware Control.

9.3.3.1 Redeploying Java EE Applications Using Fusion Middleware Control

To redeploy a Java EE application to a Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments**.
2. Select the application to redeploy.
The application home page is displayed.
3. From the Application Deployment menu, choose **Deployments**.
4. Click **Redeploy**.
The Select Application page is displayed.
5. Click **Next**.
6. In the Archive or Exploded Directory section, you can select one of the following:
 - **Archive is on the machine where this browser is running.** Enter the location of the archive or click **Browse** to find the archive file.
 - **Archive or exploded directory is on the server where Enterprise Manager is running.** Enter the location of the archive or click **Browse** to find the archive file.
7. In the Deployment Plan section, you can select one of the following:
 - **Create a new deployment plan when deployment configuration is done.**
 - **Deployment plan is on the machine where this Web browser is running.** Enter the path to the plan or click **Browse** to find the plan file.
 - **Deployment plan is on the server where Enterprise Manager is running.** Enter the path to the plan or click **Browse** to find the plan file.
8. Click **Next**.
The Application Attributes page is displayed.
9. Click **Next**.
The Deployment Wizard, Deployment Settings page is displayed.

10. On this page, you can perform common tasks before deploying your application or you can edit the deployment plan or save it to a disk. Depending on the type of application, in the Deployment Tasks section, you can:

- Configure Web modules
- Configure application security
- Configure EJB modules
- Configure persistence

See [Section 9.6](#) for detailed information about these tasks.

11. **Expand Deployment Plan.**

You can edit and save the deployment plan, if you choose. If you edit the deployment plan and change descriptor values, those changes are saved to the deployment plan. In addition, the following configurations are saved to the deployment plan:

- Application attributes
- Web module configuration
- EJB configuration

Application attributes related to MDS are stored in the file `adf-config.xml`.
Application security attributes are stored in `weblogic-application.xml`.

Fusion Middleware Control updates the relevant files and repackages the `.ear` file.

12. **Click Redeploy.**

Processing messages are displayed.

13. When the operation completes, click **Close**.

To redeploy an application to a cluster, select the cluster. Then, from the target's menu, select **Application Deployment**, then **Redeploy**.

9.3.3.2 Redeploying Java EE Applications Using WLST

You can redeploy an application using the WLST command line. To redeploy a Java EE application when WLST is connected to the Administration Server, you use the WLST command `redeploy`, using the following format:

```
redeploy(app_name#[planpath]#[options])
```

You must invoke the `redeploy` command on the computer that hosts the Administration Server.

For example, to redeploy the application `businessApp` from all target servers:

```
redeploy('businessApp')
```

9.4 Deploying, Undeploying, and Redeploying Oracle ADF Applications

Oracle ADF is an end-to-end application framework that builds on Java Platform, Enterprise Edition (Java EE) standards and open-source technologies to simplify and accelerate implementing service-oriented applications.

You can use Fusion Middleware Control, Oracle WebLogic Server Administration Console, Oracle JDeveloper, or the command line to deploy, undeploy, or redeploy an Oracle ADF application. The following topics describe using Fusion Middleware Control, the Administration Console, and the command line to accomplish these tasks:

- [Deploying Oracle ADF Applications](#)
- [Undeploying Oracle ADF Applications](#)
- [Redeploying Oracle ADF Applications](#)

See *Developing Fusion Web Applications with Oracle Application Development Framework* for information on developing ADF applications and for deploying them using Oracle JDeveloper

9.4.1 Deploying Oracle ADF Applications

You can deploy an application to a WebLogic Server Managed Server instance or a cluster. This section describes how to deploy an application to a Managed Server. This example assumes that you have created an .ear file containing the ADF application.

9.4.1.1 Deploying ADF Applications Using Fusion Middleware Control

To deploy an Oracle ADF application using Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the server in which you want to deploy the application.
The server home page is displayed.
3. From the WebLogic Server menu, choose **Control**, then **Deployments**.
The Deployments page is displayed.
4. Click **Deploy**.
The Select Archive page is displayed.
5. In the Archive or Exploded Directory section, you can select one of the following:
 - **Archive is on the machine where this browser is running.** Enter the location of the archive or click **Browse** to find the archive file.
 - **Archive or exploded directory is on the server where Enterprise Manager is running.** Enter the location of the archive or click **Browse** to find the archive file.
6. In the Deployment Plan section, you can select one of the following:
 - **Create a new deployment plan when deployment configuration is done.**
 - **Deployment plan is on the machine where this Web browser is running.** Enter the path to the plan.
 - **Deployment plan is on the server where Enterprise Manager is running.** Enter the path to the plan.
7. In the Deployment Type section, you can select one of the following:
 - **Deploy this archive or exploded directory as an application**
 - **Deploy this archive or exploded directory as a library**
8. Click **Next**.
The Select Target page is displayed.
9. Select the target to which you want to deploy the application.
You can select a cluster, one or more Managed Servers in the cluster, or a Managed Server that is not in a cluster.

10. Click Next.

The Application Attributes page is displayed, as shown in the following figure:

The screenshot shows the 'Deploy Java EE Application : Application Attributes' dialog box. The 'Application Name' field contains 'mdsappdb1'. The 'Context Root of Web Modules' table has the following data:

Web Module	Context Root
mdsappdbweb.war	mdsappdbweb

The 'Target Metadata Repository' section displays a warning message: "The metadata repository 'mds-appDBRepos (Database)' specified in this application is not a registered repository in this domain. Select a registered repository." Below this, the 'Repository Name' is 'mds-appDBRepos', the 'Repository Type' is 'Database', and the 'Partition' is 'partition1'.

The 'Distribution' section has three radio buttons:

- Install and start application (servicing all requests)
- Install and start application in administration mode (servicing only administration requests)
- Install only. Do not start.

11. In the Application Attributes section, for **Application Name**, enter the application name.
12. In the Context Root of Web Modules section, if the Web module does not have the context root configured in the application.xml file, you can specify the context root for your application. The context root is the URI for the Web module. Each Web module or EJB module that contains Web services may have a context root.
13. In the Target Metadata Repository section, you can choose the repository and partition for this application. If the partition name is not specified in the adf-config.xml file, the application name plus the version is used as the default partition name. This ensures that the partition used is unique in the domain so that the metadata for different applications are not accidentally imported into the same repository partition and overwrite each other. Typically, each application's metadata is deployed to its own partition.
 - To change the repository, click the icon next to the **Repository Name**. In the Metadata Repositories dialog box, select the repository and click **OK**.
 - To change the partition, enter the partition name in **Partition Name**. Oracle recommends that you create a new partition for each application. If you enter a name of a partition that does not exist, the partition is created.

The adf-config.xml file in the .ear file is updated with the new information.

If the partition or repository specified in the application is not valid in the domain, Fusion Middleware Control displays a message.

14. If the application's adf-config.xml file archive contains MDS configuration for an MDS shared repository, the Shared Metadata Repository section is displayed. In

this section, you can choose the repository and partition for this application. If the partition or repository specified in the application is not valid in the domain, Fusion Middleware Control displays a message.

If you change the repository or partition, the `adf-config.xml` file in the `.ear` file is updated with the new information.

15. In the Distribution section, you can select one of the following:
 - **Install and start application (servicing all requests)**
 - **Install and start application in admin mode (servicing only administration requests)**
 - **Install only. Do not start.**
16. You can expand Other Options. See [Section 9.3.1](#) for a description of those options.
17. Click **Next**.

The Deployment Wizard, Deployment Settings page is displayed.

18. On this page, you can perform common tasks before deploying your application or you can edit the deployment plan or save it to a disk. Depending on the type of application, in the Deployment Tasks section, you can:
 - **Configure Web modules:** Click **Go to Task** in the Configure Web Modules row. The Configure Web Modules page is displayed. Click **Configure General Properties** to view and edit the general configuration for the Web Module or **Map Resource References** to map the resource references.

For example, you can change the session invalidation interval or the maximum age of session cookies.
 - **Configure EJB modules:** Click **Go to Task** in the Configure EJB modules row to set standard EJB deployment descriptor properties. The Configure EJB Modules page is displayed. Click **Configure EJB Properties** to view and edit the general configuration for the EJBs or **Map Resource References** to map the resource preferences.

For example, you can configure the maximum number of beans in the free pool or the network access point.
 - **Configure application security:** Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed, as described in [Section 9.6](#).
 - **Configure persistence:** Click **Go to Task** in the Configure Persistence row to configure Java Persistent API (JPA) persistence units.
 - **Configure ADF Connections:** To modify the ADF connections, click **Go to Task** in the Configure ADF Connections row. The Configure ADF Connections page is displayed, showing the current connection information. To modify a connection type, click the **Edit** icon for a particular row. For example, you can modify the connection information for an external application. For more information about ADF connections, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

For more information about these options, see [Section 9.6](#).

19. Expand **Deployment Plan**.

You can edit and save the deployment plan, if you choose.
20. Click **Deploy**.

Fusion Middleware Control displays processing messages.

21. When the deployment is completed, click **Close**.

9.4.1.2 Deploying ADF Applications Using WLST or the Administration Console

You can deploy an ADF application using the WLST command line or the Oracle WebLogic Server Administration Console.

Take the following steps:

1. If your application uses an MDS Repository, you must configure the application archive (.ear) file before you deploy your application. You must provide the repository information for the deploy target repository and any shared metadata repositories using the WLST `getMDSArchiveConfig` command. The repository specified must already be registered with the domain before deploying the application. The following examples show how to use this command to get the `MDSArchiveConfig` and call the `setAppMetadataRepository` method to set the deploy target repository. Otherwise, your application will fail to start.

```
wls:/offline>#archive##getMDSArchiveConfig(fromLocation='/tmp/App1.ear')
wls:/offline>#archive.setAppMetadataRepository(repository='AppRepos1',
#####partition='partition1',#type='DB',#jndi='mds-jndi1')
```

The operation places the changes in the MDS configuration portion of the `adf-config.xml` file in the archive file.

2. Save the changes to the original .ear file, using the following command:

```
wls:/offline>#archive.save()
```

3. Deploy the application.

To deploy an application when WLST is connected to the Administration Server, you use the WLST command `deploy`, using the following format:

```
deploy(app_name,#path#[,targets]#[,stageMode]#[,planPath]#[,options])
```

You must invoke the `deploy` command on the computer that hosts the Administration Server.

For example, to deploy the application `myApp`:

```
deploy("myApp", "/scratch/applications/myApp", #targets='myserver', #
timeout=120000))
```

See Also:

- "Deployment Tools" in *Deploying Applications to Oracle WebLogic Server* for more information about using WLST to deploy applications
- *WLST Command Reference for WebLogic Server*

To deploy the application using the Oracle WebLogic Server Administration Console:

1. If you have not already done so, in the Change Center of the Administration Console, click **Lock & Edit**.
2. In the left pane of the Administration Console, select **Deployments**.
3. In the right pane, click **Install**.

9.4.2 Undeploying Oracle ADF Applications

To undeploy an Oracle ADF application using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments**.
2. Select the application to undeploy.
The application home page is displayed.
3. From the Domain Application Deployment menu, choose **Deployments**.
4. Select the application.
5. In Confirmation page, click **Undeploy**.
Processing messages are displayed.
6. When the operation completes, click **Close**.

Alternatively, you can navigate to the domain, Managed Server, or cluster. Then, from the target's menu, choose **Application Deployment**, then **Undeploy**. In the Select Application page, select the application you want to undeploy.

Note that when you undeploy an application, documents stored in the MDS partition are not deleted.

9.4.3 Redeploying Oracle ADF Applications

When you redeploy an application, if the application contains a Metadata Archive (MAR), the contents of the MAR is imported to the application's metadata repository only if the MAR is changed. If the MAR is unchanged from previous deployment of the application, it is ignored.

If you are redeploying a non-versioned application or a versioned application with the same version, note the following:

- The file name and path for the archive you are redeploying must be identical to the file name and path you used when you initially deployed the application.
For example, if the file name and path of the original application was `/dua0/staging/myApp.ear`, then the revised application must be `/dua0/staging/myApp.ear`.
- If you initially deployed the application using the Oracle WebLogic Server Administration Console or WLST or other management tools other than Fusion Middleware Control, then you cannot redeploy the application using Fusion Middleware Control.

To redeploy an Oracle ADF application using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments**.
2. Select the application to redeploy.
The application home page is displayed.
3. From the Domain Application Deployment menu, choose **Deployments**.
4. Click **Redeploy**.
5. Click **Next**.
The Select Archive page is displayed.
6. In the Archive or Exploded Directory section, you can select one of the following:

- **Archive is on the machine where this browser is running.** Enter the location of the archive or click **Browse** to find the archive file.
 - **Archive or exploded directory is on the server where Enterprise Manager is running.** Enter the location of the archive or click **Browse** to find the archive file.
7. In the Deployment Plan section, you can select one of the following:
 - **Create a new deployment plan when deployment configuration is done.**
 - **Deployment plan is on the machine where this web browser is running.** Enter the path to the plan.
 - **Deployment plan is on the server where Enterprise Manager is running.** Enter the path to the plan.
 8. Click **Next**.

The Application Attributes page is displayed.
 9. In the Application Attributes section, for **Application Name**, enter the application name.
 10. In the Context Root of Web Modules section, if the Web module does not have the context root configured in the application.xml file, you can specify the context root for your application. The context root is the URI for the Web module. Each Web module or EJB module that contains Web services may have a context root.
 11. The Target Metadata Repository section is displayed. In this section, you can choose the repository and partition for this application:
 - To change the repository, click the icon next to the **Repository Name**. In the Metadata Repositories dialog box, select the repository and click **OK**.
 - To change the partition, enter the partition name in **Partition Name**. Oracle recommends that you create a new partition for each application. If you enter a name of a partition that does not exist, the partition is created.
 12. If the application's adf-config.xml file archive contains MDS configuration for an MDS shared repository, the Shared Metadata Repository section is displayed. In this section, you can choose the repository and partition for this application.
 13. Click **Next**.

The Deployment Settings page is displayed.
 14. On this page, you can perform common tasks before deploying your application or you can edit the deployment plan or save it to a disk. In the Deployment Tasks section, you can:
 - Configure Web modules
 - Configure application security
 - Configure persistence

See [Section 9.6](#) for detailed information about these options.
 15. Expand **Deployment Plan**.

You can edit and save the deployment plan, if you choose.
 16. Click **Deploy**.

Fusion Middleware Control displays processing messages.
 17. When the deployment is completed, click **Close**.

18. In the Confirmation page, click **Redeploy**.

9.5 Managing Deployment Plans

A **deployment plan** is a client-side aggregation of all the configuration data needed to deploy an archive into Oracle WebLogic Server. A deployment plan allows you to easily deploy or redeploy an application using a saved set of configuration settings.

A new deployment plan is created by default if you do not apply an existing deployment plan to an application at the time of deployment, as described in [Section 9.3.1](#). Once created, you can save a deployment plan as a file and reuse it for redeploying the application or for deploying other applications.

However, if you change the configuration of an application after it is deployed (for example, if you modify the MDS configuration of an application), then any existing deployment plans you saved no longer represent the configuration settings of the deployed application.

In such a situation, you can fetch a new deployment plan that more closely represents the configuration of the deployed application.

To fetch the deployment plan of an application that is currently deployed:

1. From the **WebLogic Domain** menu, choose **Deployments**.
2. Select an application from the list of currently deployed applications.
3. Click **Fetch Deployment Plan**.

The Fetch Deployment Plan page is displayed.

4. Select a location where you want to save the deployment plan, and click **Fetch**.

You can save the plan to the computer where the Web browser is running or to the computer where Fusion Middleware Control is running.

5. In the resulting dialog box, specify a directory location for the saved deployment plan.

You can now use this deployment plan to later deploy or redeploy the application using the configuration currently in use by the application.

Alternatively, you can edit a deployment plan on the Deployment Settings page of the Application Deployment wizard.

9.6 About the Common Deployment Tasks in Fusion Middleware Control

When you deploy an application using Fusion Middleware Control, you can use the Deployment Settings page of the Deployment wizard to perform specific deployment configuration tasks before the application is deployed.

The following describes the deployment tasks that can appear on the Deployment Settings page, depending on the type of application you are deploying.

Configure Web modules

This deployment task is available when you are deploying any application that includes a Web module. In most cases, this means the application contains a Web application deployment descriptor (`web.xml` or `weblogic.xml`); however, a Web module can also be identified by annotations in the Java code of the application.

You can use this deployment task to set standard Web application deployment descriptor properties, such as:

- Session validation interval
- Maximum age of session cookies

Configure EJBs

This deployment task is available for any application that includes an EJB module. In most cases, this means the application contains an EJB deployment descriptor (ejb-jar.xml or weblogic-ehb-jar.xml); however, an EJB module can also be identified by annotations in the Java code of the application.

You can use this deployment task to set standard EJB deployment descriptor properties, such as:

- The maximum number of beans in the free pool
- The EJB network access point

Configure Application Security

This deployment task is available for all application types. However, the options available when you select this task vary depending on the existence of the following files in the application:

- jazn-data.xml

If the jazn-data.xml file exists in the application, then you can:

- Append, overwrite, or ignore policy migration.
 - * If you are deploying the application for the first time, then select **Append**.
 - * If the application was previously deployed and the application authorization policy exists, then select **Append**, or select **Ignore** to keep the application authorization policy.
 - * To overwrite the previous policy, then select **Overwrite**.
- Specify the Application stripe ID, if the stripe ID is inconsistent with the one defined in the migration options.
- Specify that policies are removed when the application is undeployed.

- cwallet.sso

If an cwallet.sso file exists in the application, then you can set additional application credential migration options.

If the application contains both files, the page displays both sections.

For more information about the settings available when you select the Configure Application Security deployment task, see "Deploying Java EE and Oracle ADF Applications with Fusion Middleware Control" in *Securing Applications with Oracle Platform Security Services*.

If neither of these files exists in the application, then you can use this task to determine how user roles and policies will be defined when the application is deployed. For example, you can choose to use only the roles and policies defined in the deployment descriptors, or you can choose to use only the roles and policies defined on the server. The Configure Application Security page displays the following options:

- **Deployment Descriptors Only:** Use only roles and policies that are defined in the deployment descriptors.
- **Custom Roles:** Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.

- **Custom Roles and Policies:** Use only roles and policies that are defined in the Administration Console.
- **Advanced:** Use a custom model that you have configured on the realm's configuration page.

Configure persistence

This deployment task is available for applications that contain one or more persistence.xml files. Using this task, you can configure the Java Persistent API (JPA) persistence units for the application.

You can view details about each persistence unit and define a Java Transaction API (JTA) data source or non-JTA data source for each persistence unit.

Configuring the data sources for persistence units can be useful for applications that take advantage of Oracle TopLink. For more information, refer to the *Solutions Guide for Oracle TopLink*.

For more information about how persistence units and the persistence.xml file can be used in Java EE applications, refer to the definition of Persistence Units in the Java EE 5 Tutorial at the following Web site:

<http://download.oracle.com/javasee/5/tutorial/doc/bnbqw.html#bnbrj>

Configure ADF connections

This deployment task is available for applications that use ADF connections. You can modify the connection information for an external application. For more information about ADF connections, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

9.7 Changing MDS Configuration Attributes for Deployed Applications

If your application uses an MDS Repository, you can modify configuration attributes after the application is deployed. To view or modify the attributes, you can use the System MBean Browser or WLST.

Note: Changes to the configuration persist in MDS as customizations. Because these persist as customizations:

- Any changes made to the configuration are retained across application deployments. For example, assume that an application has an ExternalChangeDetectionInterval configuration attribute value set to 40 seconds through Oracle JDeveloper. If you change the ExternalChangeDetectionInterval configuration attribute to 50 seconds, and you redeploy the application, the value of the attribute remains at 50 seconds.
 - In a cluster, because all instances of the deployed application point to the same MDS Repository partition, all instances of the application use the same value. If a configuration attribute has been changed for one application instance, all instances of that application in a cluster use the changed value.
-
-

The following topics describe how you can change the MDS configuration attributes:

- [Changing the MDS Configuration Attributes Using Fusion Middleware Control](#)
- [Changing the MDS Configuration Using WLST](#)

- [Restoring the Original MDS Configuration for an Application](#)

9.7.1 Changing the MDS Configuration Attributes Using Fusion Middleware Control

To change the MDS configuration attributes of an application, take the following steps:

1. Navigate to the application's home page by expanding **Application Deployments**. Then, select an application.

The application's home page is displayed.

2. From the Application Deployment menu, choose **System MBean Browser**.

The System MBean Browser page is displayed.

3. Expand **Application Defined MBeans**, then **oracle.adf.share.config**, then **Server: name**, then **Application: name**, then **ADFConfig**, then **ADFConfig**, and **ADFConfig**.

4. Select **MDSAppConfig**.

The Application Defined MBeans page is displayed, as shown in the following figure:

Name	Description
1 AppMetadataRepositoryInfo	Metadata repository partition where the a
2 AutoPurgeTimeToLive	Automatically purge versions of metadata given time interval specified in seconds.
3 ConfigMBean	If true, it indicates that this MBean is a Co
4 DeployTargetRepository	The repository where the application's met
5 eventProvider	If true, it indicates that this MBean is an e JSR-77.
6 eventTypes	All the event's types emitted by this MBea
7 ExternalChangeDetection	Enables the application to detect applicabl performed external to the application.
8 ExternalChangeDetectionInterval	The maximum time interval in seconds with detect external metadata changes. This p ExternalChangeDetection is enabled.
9 MaximumCacheSize	The maximum metadata cache size limit in
10 objectName	The MBean's unique JMX name
11 ReadOnly	If true, it indicates that this MBean is a rea
12 ReadOnlyMode	Switches the application into read only mo will be made.
13 RemoteNotifications	Enables distributed remote notifications of changes. This parameter is only valid if Ext enabled.
14 RestartNeeded	Indicates whether a restart is needed.
15 RetryConnection	Enables the application to retry to connect after connection failure.
16 SharedMetadataRepositoryInfo	Shared metadata repositior partition(s) ref
17 stateManageable	If true, it indicates that this MBean provid capabilities as defined by JSR-77.
18 statisticsProvider	If true, it indicates that this MBean is a sta JSR-77.
19 SystemMBean	If true, it indicates that this MBean is a Sy
20 Visible	If true, it indicates that this MBean is visib

5. You can view the description and values for the attributes.

[Table 9–2](#) describes the configuration attributes that are specific to MDS. Note that other attributes, such as ConfigMBean appear in the browser, but these are generic attributes for all MBeans.

Table 9–2 MDS Configuration Attributes for Deployed Applications

Attribute	Description
AppMetadataRepositoryInfo	Read only. Describes the metadata repository partition where the application is deployed.
AutoPurgeTimeToLive	Automatically purge versions of metadata documents older than the given time interval, specified in seconds. Any unlabeled versions older than this time interval are automatically purged on any subsequent update from this application. If the value is not set, versions are not automatically purged.
ConfigMBean	If true, indicates that this MBean is a Config MBean.
DeployTargetRepository	The name of the target repository configured for the application.
eventProvider	If true, it indicates that this MBean is an event provider as defined by JSR-77.
eventTypes	All the event's types emitted by this MBean.
ExternalChangeDetection	<p>Specifies that the MDS Repository is polled to determine if any metadata changes have been performed on other cluster nodes or by other applications. If changes are detected, notifications are sent to applications that share the repository.</p> <p>Multiple applications can share metadata that is deployed to a shared repository. Changes performed by one application to this shared metadata can be detected by the other application. To do this, both the applications should configure the shared repository as part of their application configuration.</p> <p>If the MDS Repository is being used by more than one application in the same JVM, then MDS polls for changes if any of those applications have ExternalChangeDetection set to true.</p> <p>This attribute should only be set to false if the application metadata is never updated or if it is used only by this application and on a single server node.</p> <p>This attribute is applicable only to database-based repositories. The default is true.</p>
ExternalChangeDetectionInterval	<p>The maximum time interval, in seconds, to poll the MDS Repository to determine if there are external metadata changes. This attribute is only valid if ExternalChangeDetection is enabled.</p> <p>If the MDS Repository is shared and being used by more than one application in the same JVM, MDS uses the lowest of the values specified in the different applications for this attribute. As a result, changing the value of this parameter in one application only has an effect if the new value is lower than any values specified in the other applications.</p> <p>The default is 30 seconds.</p>
MaximumCacheSize	The maximum metadata cache size limit, in kilobytes. If the value is 0, caching is disabled. If no value is specified, there is no cache limit. In this case, cached data is stored indefinitely.
objectName	All the event's types emitted by this MBean.
ReadOnly	If true, it indicates that this MBean is a read-only MBean.

Table 9–2 (Cont.) MDS Configuration Attributes for Deployed Applications

Attribute	Description
ReadOnlyMode	Changes the application to read-only mode, so that no updates can be made to the application's repository partition, including configuration and application metadata.
RemoteNotifications	Enables distributed remote notifications of applicable metadata changes. This parameter is valid only if ExternalChangeDetection is enabled.
RestartNeeded	Enables distributed remote notifications of applicable metadata changes. This parameter is only valid if ExternalChangeDetection is enabled.
RetryConnection	Enables the application to retry the connection to the metadata repository after connection failure.
SharedMetadataRepositoryInfo	Read only. Specifies the MDS Repository partition used by the application. Note that an application can use more than one shared metadata repository.
stateManageable	If true, it indicates that this MBean provides State Management capabilities as defined by JSR-77.
statisticsProvider	If true, it indicates that this MBean is a statistic provider as defined by JSR-77.
SystemMBean	If true, it indicates that this MBean is a System MBean.
Visible	If true, it indicates that this MBean is visible to the current user.

6. To view or modify an attribute, select the attribute.
The attribute page is displayed.
7. If the attribute is not read-only, you can change the values. For example, for `AutoPurgeTimeToLive`, you can change the interval, by entering a new value in **Value**.
8. Click **Apply**.
9. Navigate up to `ADFConfig` (the parent of `MDSAppConfig`) and select it.
10. In the Operations tab, click **Save**.
11. Click **Invoke**.

9.7.2 Changing the MDS Configuration Using WLST

You can change the MDS configuration of an application using WLST. The following example shows a WLST script that reads and then sets the `ReadOnlyMode` attribute:

```
"""
Getting#ReadOnlyMode#Attribute#from#MBean
"""
connect('username', 'password', 'hostname:port')
application#=#'application_name'
attribute#=#'ReadOnlyMode'
beanName#=#'oracle.adf.share.config:ApplicationName='+#application#
+',name=MDSAppConfig,type=ADFConfig,Application='+#application#
+',ADFConfig=ADFConfig,*'

beanObjectName#=#ObjectName(beanName)
beans#=#mbs.queryMBeans(beanObjectName, #None)
```

```

bean#=#beans.iterator().next().getObjectNames()
custom()
value#=#mbs.getAttribute(bean,#attribute)
print#value

"""
Setting#ReadOnlyMode#Attribute#from#MBean
"""
attr#=#Attribute(attribute,#Boolean(0))##
mbs.setAttribute(bean,attr)
value#=#mbs.getAttribute(bean,#attribute)
print#value

"""
Saving#the#Changes.#This#is#required#to#persist#the#changes.
"""

adfConfigName#=#'oracle.adf.share.config:ApplicationName='+#application##
',name=ADFConfig,type=ADFConfig,Application='+#application##','*'
adfConfigObjectName#=#ObjectName(adfConfigName)
adfConfigMBeans#=#mbs.queryMBeans(adfConfigObjectName,#None)
adfConfigMBean#=#adfConfigMBeans.iterator().next().getObjectNames()
mbs.invoke(adfConfigMBean,#'save',#None,#None)

```

9.7.3 Restoring the Original MDS Configuration for an Application

To restore the original MDS configuration for an application:

1. Navigate to the application's home page by expanding **Application Deployments**. Then, select an application.

The application's home page is displayed.

2. From the Application Deployment menu, choose **System MBean Browser**.

The System MBean Browser page is displayed.

3. Expand **Application Defined MBeans**, then **oracle.adf.share.config**, then **Server: name**, then **Application: name**, then **ADFConfig**, and then **ADFConfig**.

4. Select the Operations tab.

5. Select **RestoreToOriginalConfiguration**.

The Operation: restoreToOriginalConfiguration page is displayed.

6. Click **Invoke**.

Use this operation with caution. It causes all changes made to the original adf-config.xml file to be discarded. The adf-config.xml is restored to the base document.

Part V

Monitoring Oracle Fusion Middleware

This part provides information about how to find information about the cause of an error and its corrective action, to view and manage log files to assist in monitoring system activity and to diagnose problems and how to monitor Oracle Fusion Middleware.

Part V contains the following chapters:

- [Chapter 10, "Monitoring Oracle Fusion Middleware"](#)
- [Chapter 11, "Managing Log Files and Diagnostic Data"](#)
- [Chapter 12, "Diagnosing Problems"](#)

Monitoring Oracle Fusion Middleware

This chapter describes how to monitor Oracle Fusion Middleware using Fusion Middleware Control, Oracle WebLogic Server Administration Console, and the command line.

It describes the following sections:

- [Monitoring the Status of Oracle Fusion Middleware](#)
- [Viewing the Performance of Oracle Fusion Middleware](#)
- [Viewing the Routing Topology](#)

10.1 Monitoring the Status of Oracle Fusion Middleware

Monitoring the health of your Oracle Fusion Middleware environment and ensuring that it performs optimally is an important task for the administrator.

Oracle Fusion Middleware provides the following methods for monitoring the status of your environment:

- **Fusion Middleware Control:** You can monitor the status of Oracle WebLogic Server domains, clusters, servers, Java components, system components, and applications. Navigate to the entity's home page, for example, to the home page for an Oracle HTTP Server instance.
- **Oracle WebLogic Server Administration Console:** You can monitor the status of Oracle WebLogic Server domains, clusters, servers, Java components, and applications. From the Administration Console, navigate to the entity's page. See "Overview of the Administration Console" in *Understanding Oracle WebLogic Server* for information on monitoring using the console.
- **The command line:** You can monitor the status of your environment using the WLST `state` command.

To monitor the status of Java components, use the WLST `state` command, with the following format:

```
state(name, #type)
```

For example, to get the status of the Managed Server `server1`, use the following command:

```
wls:/mydomain/serverConfig>#state('server1','Server')
Current#state#of#"server1":#SUSPENDED
```

To monitor the status of system components, use the WLST `state` command, with the following format:

```
state('component_name']]
```

For example, to view the status ohs1, use the following command:

```
state('ohs1']]
```

Most of the monitoring tasks in this chapter describe how to monitor using Fusion Middleware Control or the command line.

The following topics provide more detail:

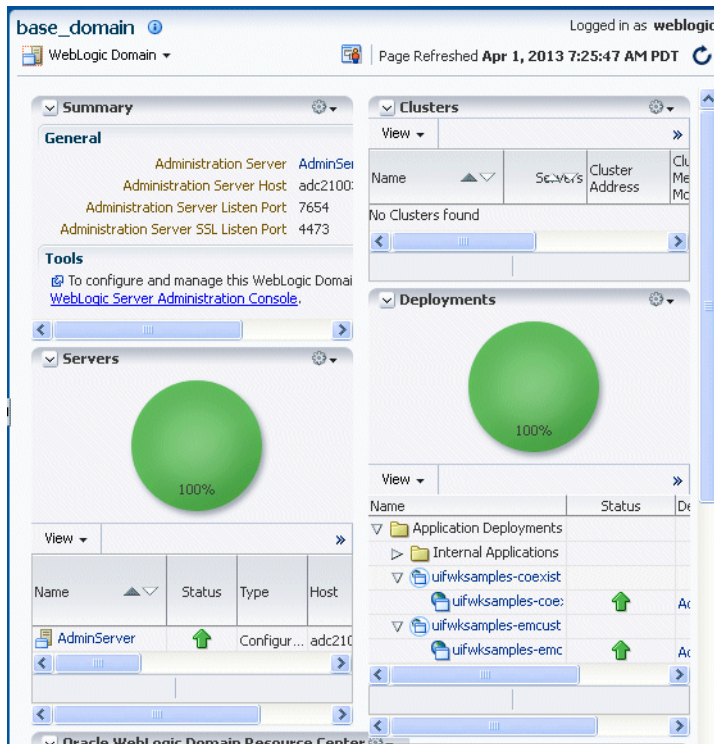
- [Monitoring an Oracle WebLogic Server Domain](#)
- [Monitoring an Oracle WebLogic Server Administration or Managed Server](#)
- [Monitoring a Cluster](#)
- [Monitoring a System Component](#)
- [Monitoring Java EE Applications](#)
- [Monitoring ADF Applications](#)
- [Monitoring Applications Deployed to a Cluster](#)

10.1.1 Monitoring an Oracle WebLogic Server Domain

You can view the status of a domain, including the servers, clusters, and deployments in the domain from the domain home page of Fusion Middleware Control:

1. From the WebLogic Domain menu, select **Home**.

The domain home page is displayed, as shown in the following figure:



This page shows the following:

- A general summary of the domain, along with a link to the Oracle WebLogic Server Administration Console

- Information about the servers, both the Administration Server and the Managed Servers, in the domain
- Information about the clusters in the domain
- Information about the deployments in the domain
- A Resource Center, which provides links to more information

For information about monitoring an Oracle WebLogic Server domain using the Oracle WebLogic Server Administration Console, see "Overview of the Administration Console" in the *Understanding Oracle WebLogic Server*. The Administration Console provides details about the health and performance of the domain.

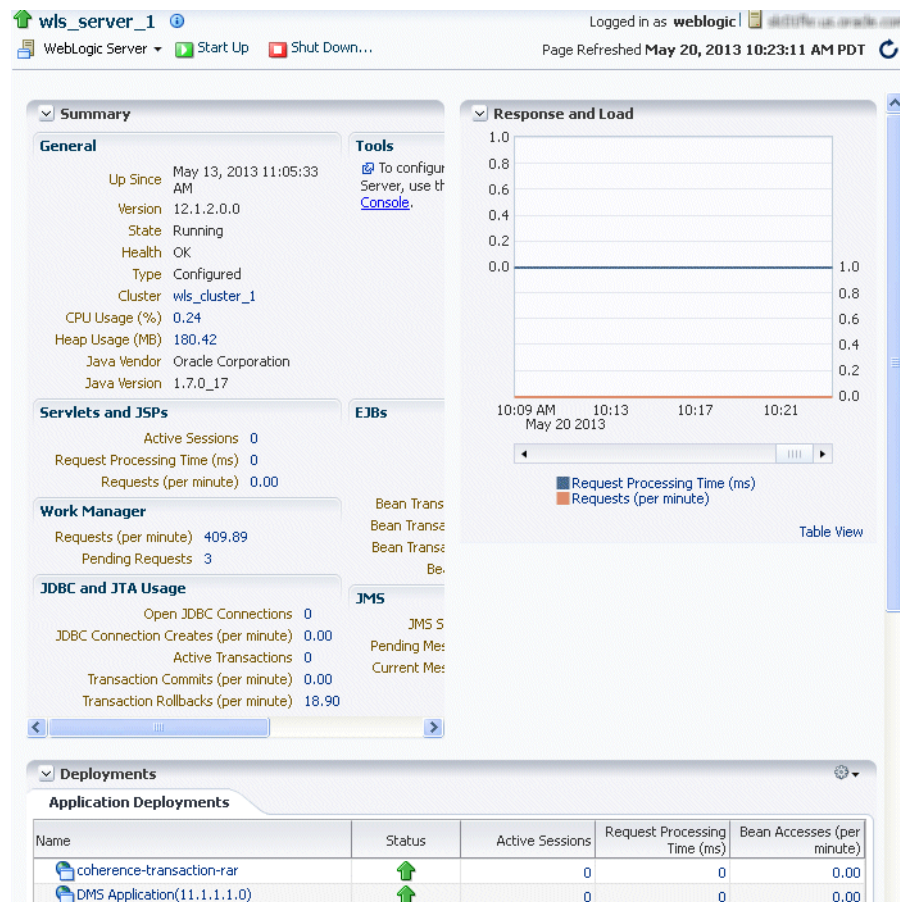
10.1.2 Monitoring an Oracle WebLogic Server Administration or Managed Server

You can view the status of a WebLogic Server Administration Server or Managed Server in Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the server.

The server home page is displayed.

The following figure shows the home page for a Managed Server:



This page shows the following:

- A general summary of the server, including its state, and information about the servlets, JSPs, and EJBs running in the server

- Response and load
- Information about the applications deployed to the server

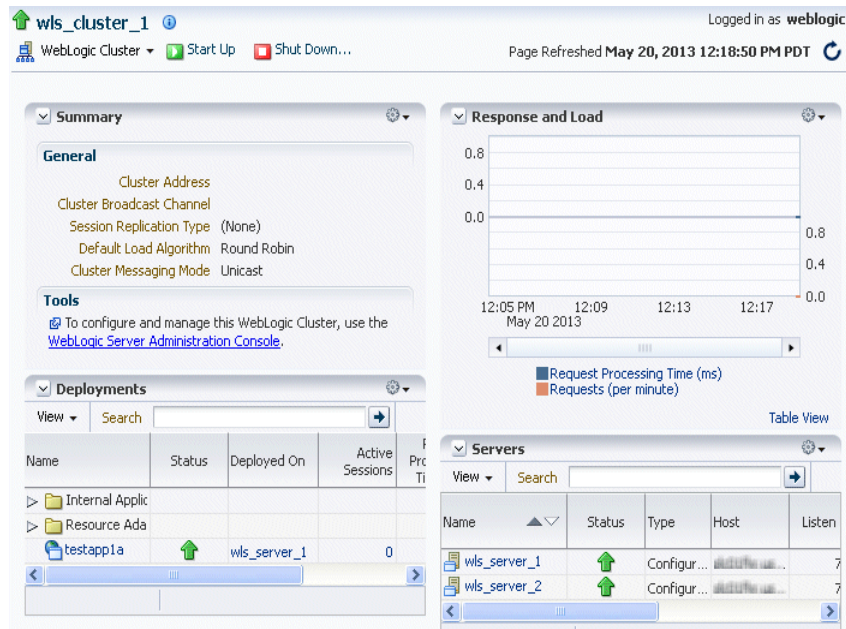
For information about monitoring servers using the Oracle WebLogic Server Administration Console, see "Overview of the Administration Console" in *Understanding Oracle WebLogic Server*. The Administration Console provides details about the health and performance of the server.

10.1.3 Monitoring a Cluster

You can view the status of a cluster, including the servers and deployments in the cluster using Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the cluster.

The cluster page is displayed, as shown in the following figure:



This page shows the following:

- A general summary of the cluster, including the broadcast channel, if appropriate, the load algorithm, and the messaging mode
- A response and load section, which shows the requests per minute and the request processing time
- A deployments section with information about the applications deployed to the cluster
- A servers section, with a table listing the servers that are part of the cluster

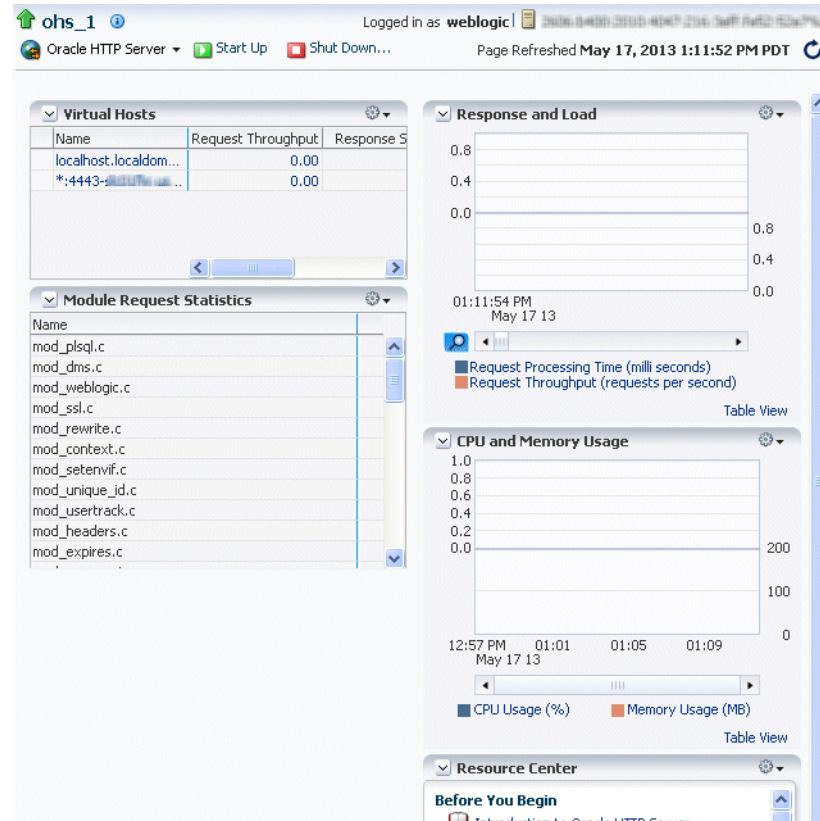
See Also: "Overview of the Administration Console" in the *Understanding Oracle WebLogic Server* for information about monitoring a cluster using Oracle WebLogic Server Administration Console. The Administration Console provides details about the health and performance of the cluster.

10.1.4 Monitoring a System Component

To monitor a system component, such as Oracle HTTP Server:

1. From the navigation pane, expand **Web Tier**.
2. Select the component, such as ohs1.

The component home page is displayed, as shown in the following figure:



This page shows the following:

- A response and load section, which shows the requests per second and the request processing time
- CPU and memory usage
- The virtual hosts, with their names, request throughput and response size.
- Module request statistics, with a list of modules and the throughput for each.: A table listing the processing time for each module
- A Resource Center with links to relevant documentation

10.1.5 Monitoring Java EE Applications

To monitor a Java EE application using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments**, then select the application to monitor.

The application's home page is displayed.

- In this page, you can view a summary of the application's status, entry points to the application, Web services and modules associated with the application, and the response and load.

The following figure shows a portion of the application's home page:

The screenshot shows the Oracle Fusion Middleware monitoring interface for an application named 'testapp1a'. The page is titled 'testapp1a' and shows the user is logged in as 'weblogic'. The page was refreshed on May 20, 2013 at 12:26:20 PM PDT. The main content area is divided into several sections:

- Summary - General:**
 - State: Active
 - Health: OK
 - Source Path: sysman/upload/deploy/testapp1a/archive/testapp1a.ear
 - Application Type: Enterprise Application
- Deployments:** A table showing the deployment of 'testapp1a' with a status of 'Active' and an upward arrow icon.
- Modules:** A table showing the module 'testapp1-web.war' with a type of 'Web Application'.
- Web Services:** A section indicating 'No Web Services Found'.
- Data Sources:** A section indicating 'No datasources found'.

This page shows the following:

- A summary of the application, including its state, the Managed Server on which it is deployed, and information about active sessions, active requests, and request processing time
- A Deployments section, with the application and its status.
- A list of modules with the type of module for each
- Response and load, which shows the requests per minute and the request processing time
- A list of most requested servlets, JSPs, and Web services

10.1.6 Monitoring ADF Applications

To monitor an ADF application:

- From the navigation pane, expand **Application Deployments**, then select the application to monitor.

The application's home page is displayed.

The following figure shows a portion of the application's home page:

- A summary of the application, including its state, the Managed Server on which it is deployed, and information about active sessions, active requests, and request processing time
 - Deployments, which lists the servers on which the application is deployed.
 - A list of modules with the type of module for each
 - A list of data sources
2. To view health of the environment, from the **Application Deployments** menu choose **Monitoring**, then **Environment Monitoring**. The Environment Monitoring page is displayed.
 - It contains tabs for Health, Query Caching, Workload, and Coherence.

10.1.7 Monitoring Applications Deployed to a Cluster

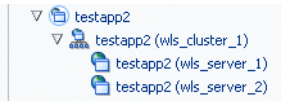
If you deploy an application to a cluster, Oracle Fusion Middleware automatically deploys the application to each Managed Server in the cluster. As a result, there is an instance of the application on each server.

There are times when you want to monitor the performance of the application on an individual server, and times when you want to monitor the overall performance of the application across all the servers in the cluster.

For example, normally, you would manage the overall performance of the application to determine if there are any performance issues affecting all users of the application, regardless of which instance users access. If you notice a performance problem, you can then drill down to a specific instance of the application to determine if the problem is affecting one or all of the application instances in the cluster.

Fusion Middleware Control provides monitoring pages for both of these scenarios:

1. From the navigation pane, expand **Application Deployments**.
 - Fusion Middleware Control lists the applications deployed in the current domain.
2. If an application has been deployed to a cluster, expand the application in the navigation pane. Fusion Middleware Control shows that it is deployed to the cluster application to indicate that it represents more than one instance of the application on the cluster:



3. Monitor the overall performance of the application on the cluster by clicking the cluster application, or monitor the performance of the application on a single server by clicking one of the application deployment instances.

10.2 Viewing the Performance of Oracle Fusion Middleware

If you encounter a problem, such as an application that is running slowly or is hanging, you can view more detailed performance information, including performance metrics for a particular target, to find out more information about the problem.

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

Note that Fusion Middleware Control provides real-time data. If you are interested in viewing historical data, consider using Oracle Enterprise Manager Grid Control.

For example, to view the performance of an Oracle WebLogic Server Managed Server:

1. From the navigation pane, expand the domain.
2. Select the server to monitor.

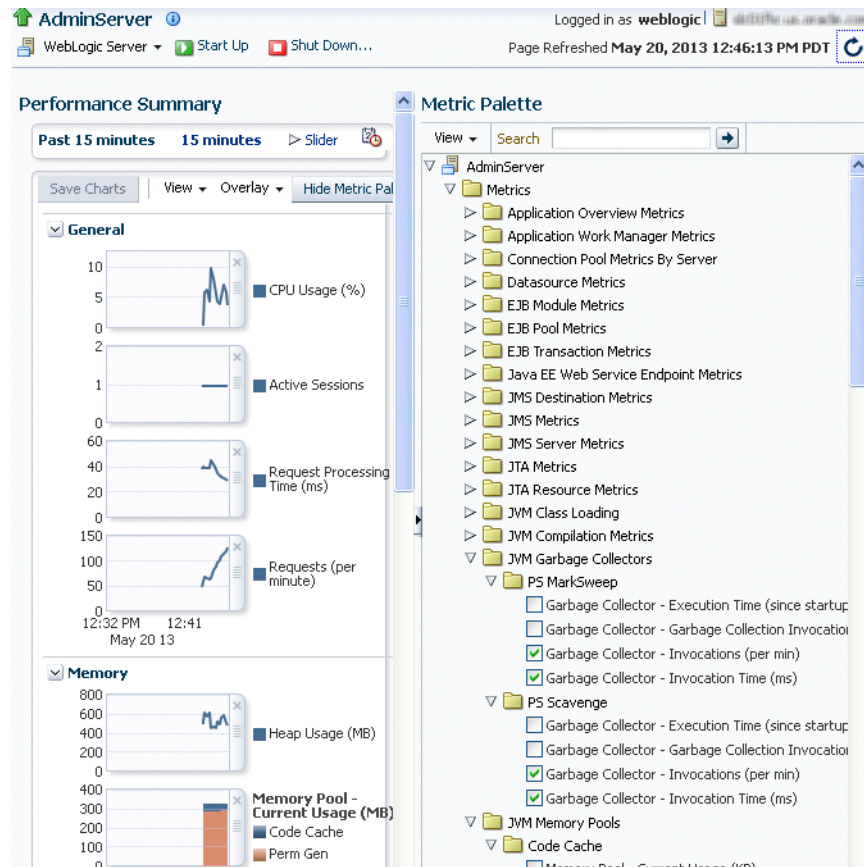
The Managed Server home page is displayed.

3. From the WebLogic Server menu, choose **Monitoring**, then **Performance Summary**.

The Performance Summary page is displayed. It shows performance metrics, as well as information about response time and request processing time for applications deployed to the Oracle WebLogic Server.

4. To see additional metrics, click **Show Metric Palette** and expand the metric categories.

The following figure shows the Performance Summary page with the Metric Palette displayed:



5. Select a metric to add it to the Performance Summary.
6. To overlay another target, click **Overlay**, and select the target. The target is added to the charts, so that you can view the performance of more than one target at a time, comparing their performance.
7. To customize the time frame shown by the charts, you can:
 - Click **Slider** to display a slider tool that lets you specify that more or less time is shown in the charts. For example, to show the past 10 minutes, instead of the past 15 minutes, slide the left slider control to the right until it displays the last 10 minutes.
 - Select the calendar and clock icon. Then, enter the **Start Time** and **End Time**. If there is no data available for those times, a confirmation message displays, explaining the timeline will be automatically adjusted to the time period for which the data is available.

You can also view the performance of a components, such as Oracle HTTP Server. Navigate to the component and select **Monitoring**, then **Performance Summary** from the dynamic target menu.

10.3 Viewing the Routing Topology

Fusion Middleware Control provides a Topology Viewer for the domain. The Topology Viewer is a graphical representation of routing relationships across components and elements of the domain. You can easily determine how requests are routed across components. For example, you can see how requests are routed from Oracle HTTP Server, to a Managed Server, to a data source.

Note: To view relationships between Oracle WebLogic Server and Oracle HTTP Server, each target must be running and show its status as Up.

The Topology Viewer enables you to easily monitor your Oracle Fusion Middleware environment. You can see which entities are up and which are down.

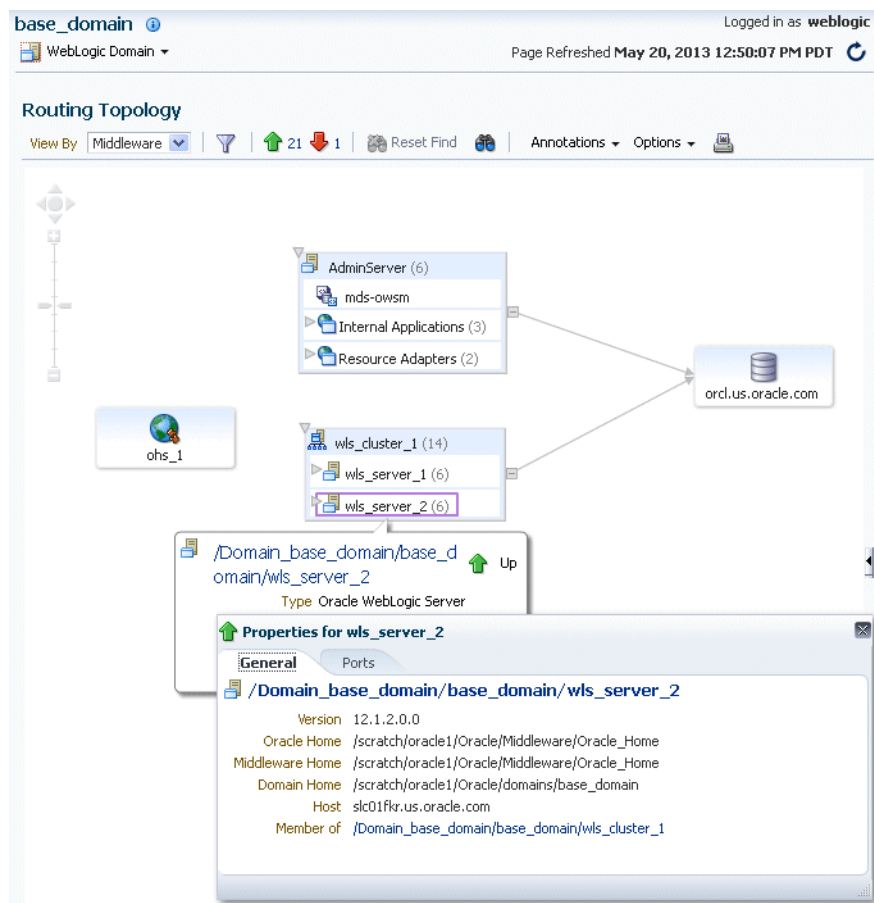
You can also print the topology.

To view the topology:

1. From the WebLogic Domain menu, click **Routing Topology**.

The Routing Topology page is displayed.

2. The following shows the Routing Topology page with information about the Managed Server:



With Topology Viewer, you can also:

- Choose how to group the routing. From the View menu, you can choose to group by Middleware, host, or application.
- View the targets by status. Click the green up arrow or the red down arrow at the top of the page. A list of the targets with the specified status is shown.
- Search for a target within the topology. This makes it easier to find a target if you have many targets. Click the **Toggle Find Toolbar**. Then, enter the name in the **Find** box, and click **Find**.

The Find results box is displayed. Click the target name to highlight the target. The topology is repositioned so you can see the target if it was not previously visible in the viewing area.

You can also specify criteria for the search. From **Find**, choose the one or more types of **Status** or one or more of **Target Type**, or both.

- Hide or show the status or metrics. From **Annotations**, click **Status** or **Metrics**.
If you select Metrics, one or more key performance metrics for the component are displayed. (You cannot change the metrics that are displayed.)
- Reposition the topology and change its orientation:
 - To change the orientation, from the Options menu, choose **Layout**, then **Left to Right** or **Top Down**.
 - To reposition the topology, click in the topology, but not on a target or route. Drag the topology to position it.
 - To change what is visible in the topology view, click the arrow in the right-hand bottom corner. Then, drag the shaded section in the navigator window, which is located in the bottom right.
- Navigate to the home page of a target. Right-click the target, and select **Home**.
- Perform operations directly on the target by right-clicking. The right-click target menu is displayed. For example, from this menu, you can start or stop an Oracle WebLogic Server or view additional performance metrics.
- View the routing relationships between components. For example, you can view the routing from Oracle HTTP Server to Oracle WebLogic Server. Clicking on the line between the two targets displays the URLs used.

Managing Log Files and Diagnostic Data

Oracle Fusion Middleware components generate log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, and access information on HTTP requests. This chapter describes how to find information about the cause of an error and its corrective action and to view and manage log files to assist in monitoring system activity and in diagnosing problems.

It contains the following sections:

- [Overview of Oracle Fusion Middleware Logging](#)
- [Understanding ODL Messages and ODL Log Files](#)
- [Viewing and Searching Log Files](#)
- [Configuring Settings for Log Files](#)
- [Correlating Messages Across Log Files and Components](#)
- [Configuring Tracing](#)

11.1 Overview of Oracle Fusion Middleware Logging

The following topics describe HTTP access logging and diagnostic logging.

- [Understanding Oracle Fusion Middleware HTTP Access Logging](#)
- [Understanding Oracle Fusion Middleware Diagnostic Logging](#)

11.1.1 Understanding Oracle Fusion Middleware HTTP Access Logging

By default, Oracle WebLogic Server is configured to use the common log format for HTTP access logs. Oracle WebLogic Server also supports the extended log format, an emerging standard defined by the draft specification from the World Wide Web Consortium (W3C).

When you install Oracle WebLogic Server with Oracle JRF, it uses the extended log format for HTTP access logs by default. The extended log format allows you to specify the type and order of information recorded about each HTTP communication.

Oracle Fusion Middleware supports the following field identifiers:

- **date**: The date at which transaction completed. The field has the format YYYY-MM-DD. All dates are specified in GMT.
- **time**: The time at which transaction completed. The field has the format HH:MM, HH:MM:SS or HH:MM:SS.S where HH is the hour in 24 hour format, MM is minutes and SS is seconds. All times are specified in GMT.

- `cs-method`: The request method, for example GET or POST. This field has type `<name>`, as defined in the W3C specification.
- `cs-url`: The full requested URI. This field has type `<uri>`, as defined in the W3C specification.
- `ctx-ecid`: The Execution Context ID (ECID). The ECID is a globally unique identifier associated with the execution of a particular request.
- `ctx-rid`: The Relationship ID (RID). The RID distinguishes the work done in one thread on one process, from work done by any other threads on this and other processes on behalf of the same request.
- `sc-status`: The status code of the response, for example (404) indicating a "File not found" status. This field has type `<integer>`, as defined in the W3C specification.

For information about the extended log format fields, see:

<http://www.w3.org/TR/WD-logfile.html>

11.1.2 Understanding Oracle Fusion Middleware Diagnostic Logging

Most Oracle Fusion Middleware components write diagnostic log files in the Oracle Diagnostic Logging (ODL) format. Log file naming and the format of the contents of log files conforms to an Oracle standard. By default, the diagnostic messages are written in text format.

ODL provides the following benefits:

- The capability to limit the total amount of diagnostic information saved. You can set the level of information saved and you can specify the maximum size of the log file and the log file directory.
- When you reach the specified size, older segment files are removed and newer segment files are saved in chronological fashion.
- Components can remain active, and do not need to be shutdown, when older diagnostic logging files are deleted.

You can view log files using Fusion Middleware Control or the WLST `displayLogs` command, or you can download log files to your local client and view them using another tool (for example, a text editor or another file viewing utility).

Note: Oracle WebLogic Server does not use the ODL format. For information about the Oracle WebLogic Server log format, see *Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

11.2 Understanding ODL Messages and ODL Log Files

Using ODL, diagnostic messages are written to log files and each message includes information, such as the time, component ID, and user.

The following example shows an ODL format error messages from Oracle HTTP Server:

```
[2013-05-13T12:31:29.0584-07:00]#[OHS]#[NOTIFICATION:16]#[OHS-9999]#[mod_weblogic.c]#[host_id:#example]#[host_addr:#nn.nnn.nn.nn]#[pid:#12789]#[tid:#46919953675776]#[user:#username#VirtualHost:#main WebLogic#Server#Plugin#version#12.1.2#<WLSPLUGINS_MAIN_LINUX.X64_130502.1731>
```

In the message, the fields map to the following attributes, which are described in [Table 11-1](#):

- Timestamp, originating: 2013-05-13T12:31:29.0584-07:00
- Organization ID: OHS
- Message Type: NOTIFICATION:16
- Component ID: mod_weblogic.c
- Host ID: host_id:#example
- Host Address: host_addr:#nn.nnn.nn.nn#
- Process ID: pid:#12789
- Thread ID: tid:#46919953675776
- User ID: userId:#username#
- Virtual Host: *VirtualHost: main*
- Message Text: "WebLogic#Server#Plugin#version#12.1.2#<WLSPLUGINS_MAIN_LINUX.X64_130502.1731>"

By default, the information is written to the log files in ODL text format. You can change the format to ODL XML format, as described in [Section 11.4.4](#).

[Table 11-1](#) describes the contents of an ODL message. For any given component, the optional attributes may not be present in the generated diagnostic messages.

Table 11-1 ODL Format Message Fields

Attribute Name	Description	Required
Timestamp, Originating (TIME)	The date and time when the message was generated. This reflects the local time zone.	Yes
Timestamp, normalized (time_norm)	The timestamp normalized for clock drift across hosts. This field is used when the diagnostic message is copied to a repository on a different host.	No
Organization ID (org_id)	The organization ID for the originating component.	No
INSTANCE_ID # (INST_ID)	The name of the instance to which the component that originated the message belongs.	No
COMPONENT ID (COMP_ID)	The ID of the component that originated the message.	Yes
MESSAGE_ID (MSG_ID)	The ID that uniquely identifies the message within the component. The ID consists of a prefix that represents the component, followed by a dash, then a 5-digit number. For example: OHS-51009	Yes
MESSAGE_TYPE # (MSG_TYPE)	The type of message. Possible values are: INCIDENT_ERROR, ERROR, WARNING, NOTIFICATION, TRACE, and UNKNOWN. See Table 11-3 for information about the message types.	Yes
MESSAGE_LEVEL (MSG_LEVEL)	The message level, represented by an integer value that qualifies the message type. Possible values are from 1 (highest severity) through 32 (lowest severity). See Table 11-3 for information about the message levels.	Yes
HOST_ID (HOST_ID)	The name of the host where the message originated.	No
HOST_NW_ADDR (HOST_ADDR)	The network address of the host where the message originated.	No
MODULE_ID (MODULE)	The ID of the module that originated the message. If the component is a single module, the component ID is listed for this attribute.	Yes

Table 11–1 (Cont.) ODL Format Message Fields

Attribute Name	Description	Required
PROCESS_ID (PID)	The process ID for the process or execution unit associated with the message.	No
THREAD_ID (TID)	The ID of the thread that generated the message.	No
USER_ID (USER)	The name of the user whose execution context generated the message.	No
ECID	The Execution Context ID (ECID), which is a global unique identifier of the execution of a particular request in which the originating component participates. You can use the ECID to correlate error messages from different components. See Section 11.5 for information about ECIDs.	Yes
RID	The relationship ID (RID), which distinguishes the work done in one thread on one process, from work done by any other threads on this and other processes, on behalf of the same request. See Section 11.5 for information about RIDs.	No
SUPPL_ATTRS	An additional list of name/value pairs which contain component-specific attributes about the event. Oracle Fusion Middleware provides the supplemental attribute DSID (Diagnostic Session ID). DSID is an ID for a user session and is used to map a set of log messages, incidents, and other diagnostic data to a user session. For example, you can see if a specific incident generated in a user session may have been preceded by earlier incidents in the same session, and could therefore be the root cause of the subsequent incident.	No
MESSAGE TEXT (TEXT)	The text of the message.	Yes
Message Arguments (arg)	A list of arguments bound with the message text.	No
Supplemental Detail	Supplemental information about the event, including more detailed information than the message text.	No

The log file location depends on the type of component:

- For most Java components, the log file location is:

```
(UNIX) #DOMAIN_HOME/servers/server_name/logs
(Windows) #DOMAIN_HOME\servers\server_name\logs
```

The default name of a log file is *server-name-diagnostic.log*.

- For system components, the default log file location is:

```
(UNIX) #DOMAIN_HOME/servers/component_name/logs
(Windows) #DOMAIN_HOME\servers\component_name\logs
```

[Table 11–2](#) shows the log file location for components of Oracle Fusion Middleware.

In the table, *DOMAIN_HOME* refers to the domain home.

Table 11–2 Log File Location

Component	Log File Location
Fusion Middleware Control	<i>DOMAIN_HOME</i> /sysman/log/emoms.log <i>DOMAIN_HOME</i> /sysman/log/emoms.trc
Oracle Application Development Framework	<i>DOMAIN_HOME</i> /servers/server_name/logs/server-name-diagnostic.log
Oracle HTTP Server	<i>DOMAIN_HOME</i> /servers/component_name/logs/*.log

Table 11–2 (Cont.) Log File Location

Component	Log File Location
Oracle Web Services Manager	<code>DOMAIN_HOME/servers/server_name/logs/owsm/msglogging</code> <code>DOMAIN_HOME/servers/server_name/logs/owsm-diagnostic.log</code>
Oracle WebLogic Server	<code>DOMAIN_HOME/servers/server_name/logs/server-name-diagnostic.log</code>
Repository Creation Utility	By default, writes to file specified in <code>RCU_LOG_LOCATION</code> . If not specified, attempts to write to the following locations: <ol style="list-style-type: none"> <code>ORACLE_HOME/rcu/log/timestamp</code> <code>/tmp/logdir.timestamp</code>

11.3 Viewing and Searching Log Files

You can view, list, and search log files across Oracle Fusion Middleware components. You can view and search log files using Fusion Middleware Control or you can download a log file to your local client and view the log files using another tool. You can also list, view, and search log files using the WLST command-line tool.

This section covers the following topics:

- [Viewing Log Files and Their Messages](#)
- [Searching Log Files](#)
- [Downloading Log Files](#)

Note the following about using the WLST commands to view the log files:

- To use the custom WLST logging commands, you must invoke the WLST script from the Oracle Common home. See [Section 2.4.2](#) for more information.
- The log viewing commands work whether you are connected or not connected to a WebLogic server. If you are not connected, you must specify the path in the `oracleInstance` parameter. You specify the domain home.
- Most of the WLST logging commands require that you are running in the `domainRuntime` tree. For example, to connect and to run in the `domainRuntime` tree, use the following WLST commands:

```
connect('username',#'password',#'localhost:port_number')
domainRuntime()
```

For more information about the commands, see "Logging Custom WLST Commands" in the *WLST Command Reference for Infrastructure Components*.

11.3.1 Viewing Log Files and Their Messages

You can view the log files using Fusion Middleware Control or WLST commands, as described in the following topics:

- [Viewing Log Files and Their Messages Using Fusion Middleware Control](#)
- [Viewing Log Files and Their Messages Using WLST](#)

11.3.1.1 Viewing Log Files and Their Messages Using Fusion Middleware Control

You can view the messages for all of the entities in a domain, an Oracle WebLogic Server, a component, or an application.

For example, to view the log files and their messages for a Managed Server:

The screenshot shows the Oracle WebLogic Server console interface. At the top, it indicates the user is logged in as 'weblogic' and the page was refreshed on May 20, 2013, at 1:49:19 PM PDT. The main content area is titled 'View Log File: wls_server_1-diagnostic.log'. It shows the file path: /scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_1/logs/wls_server_1-diagnostic.log. The log type is 'Server' and the size is 6,876.00 KB. The last modified date is May 20, 2013, at 1:49:01 PM PDT. Below this, there are search filters for 'Date Range', 'Time Interval', 'Start Date' (May 20, 2013 12:52:11), and 'End Date' (May 20, 2013 01:49:16). A search criteria dropdown is set to 'contains'. A 'View Related Messages' dropdown is also present. The main table has columns: Time, Message Type, Message ID, and Message. The first row is highlighted, showing a warning message at 12:52:11 PM PDT with ID WSM-09004 and the message 'Component auditing cannot be initialized.'. Below the table, the details for the selected message are shown, including Message ID (WSM-09004), Message Level (1), ECID (66217af9-247f-4344-94a9-14f90e75a586-0005ed5e), Relationship ID (0), Component (wls_server_1), Message (Component auditing cannot be initialized.), Module (oracle.wsm.resources.common), Host (slc01fkr), Host IP Address (10.240.82.231), User (OracleSystemUser), and Thread ID ([ACTIVE].ExecuteThread: '9' for queue: weblogic.kernel.Default (self-tuning)).

By default, the messages are sorted by time, in ascending order. You can sort the messages by any of the columns, such as Message Type, by clicking the column name.

- To view messages that are related by time or ECID, click **View Related Messages** and select **by Time** or **by ECID (Execution Context ID)**.

The Related Messages page is displayed.

11.3.1.2 Viewing Log Files and Their Messages Using WLST

You can list the log files for an Oracle WebLogic Server domain, a server, or component using the WLST `listLogs` command.

You can use this command while connected or disconnected. While connected, the default target is the Oracle WebLogic Server domain.

To list the log files, first use the `domainRuntime` command as described in [Section 11.3](#). The following describes how to list and view log files:

- To list all of the log files for the Oracle WebLogic Server `wls_server_1`, use the following command:

```
listLogs(target='wls_server_1')
file://slc01fkr/scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_1/logs/wls_server_1.log
2013-05-21#06:55:37#####500.1K#wls_server_1.log00026
2013-05-21#07:49:08#####500.1K#wls_server_1.log00027
2013-05-21#08:46:29#####500.4K#wls_server_1.log00028
2013-05-21#09:45:29#####500.4K#wls_server_1.log00029
```

```

2013-05-21#10:43:00#####500.3K#wls_server_1.log00030
2013-05-21#11:39:56#####500.3K#wls_server_1.log00031
2013-05-21#12:38:56#####500.4K#wls_server_1.log00032
2013-05-21#13:18:06#####358.1K#wls_server_1.log
#
file://slc01fkr/scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_
1/logs/wls_server_1.out
2013-05-13#11:00:05#####4M#wls_server_1.out00001
2013-05-21#13:18:06#####12.1M#wls_server_1.out
...

```

- To list the logs for a system component, use one of the following formats:

```

listLogs(target='component_name')
listLogs(target='sc:component_name')

```

For example, to list the logs for the Oracle HTTP Server ohs1, use the following command:

```
listLogs(target='ohs1')
```

- To list the logs while disconnected, you must specify the `oracleInstance` parameter, passing it the path of the domain. For example, to list the log files for the Managed Server `wls_server_1`:

```

listLogs(oracleInstance='/scratch/Oracle/config/domains/WLS_domain',
#####target='wls_server_1')

```

- To view the diagnostic messages in log files, use the WLST `displayLogs` command. This command works when you are either connected or disconnected.

For example, to view the messages generated in the last 10 minutes in the log files for the Oracle WebLogic Server domain, use the following command:

```
displayLogs(last=10)
```

```

[2013-05-21T13:30:11.892-07:00]#[wls_server_1]#[WARNING]#[WSM-09004]
#[oracle.wsm.resources.common]#[host:#slc01fkr]#[nwaddr:#10.240.82.231]#
[tid:#[ACTIVE].ExecuteThread:#'5'#for#queue:#'weblogic.kernel.Default#
(self-tuning)']#[userId:#OracleSystemUser]#
[ecid:#66217af9-247f-4344-94a9-14f90e75a586-00070b85,0]#[APP:#wsm-pm]#[TARGET:#
/base_domain/wls_server_1/wsm-pm]#
[LOG_FILE:#/scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_
1/logs/wls_server_1-diagnostic.log]#
Component#auditing#cannot#be#initialized.
[2013-05-21T13:30:11.895-07:00]#[wls_server_1]#[NOTIFICATION]#[BEA-010227]
#[EJB]#[host:#slc01fkr]#[nwaddr:#10.240.82.231]#[tid:#[ACTIVE].ExecuteThread:
#'5'#for#queue:#'weblogic.kernel.Default#(self-tuning)']#
[userId:#OracleSystemUser]#
[ecid:#66217af9-247f-4344-94a9-14f90e75a586-00070b85,0]#[TXN_ID:#
BEA1-7438ECB7CDFCAF163A9A]#
[TARGET:#/base_domain/wls_server_1]#
[LOG_FILE:#/scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_
1/logs/wls_server_1.log]#
EJB#exception#occurred#during#invocation#from#home#or#business:
#weblogic.ejb.container.internal.StatelessEJBHomeImpl@314c2224#generated
#exception:#java.lang.reflect.UndeclaredThrowableException

```

The previous command returns the messages sorted by time, in ascending order.

- To display the logs for a system component, use one of the following formats:


```
listLogs (target='component_name')
listLogs (target='sc:component_name')
```

For example, to display the log files for the Oracle HTTP Server ohs_1, use the following command:

```
displayLogs (target='sc:ohs_1')
#
```

You can search the messages by specifying particular criteria and sort the output, as described in [Section 11.3.2](#).

See "Logging Custom WLST Commands" in the *WLST Command Reference for Infrastructure Components* for more information about the `listLogs` and `displayLogs` commands.

11.3.2 Searching Log Files

You can search for diagnostic messages by time, type of message, and certain log file attributes by using Fusion Middleware Control or WLST commands, as described in the following topics:

- [Searching Log Files Using Fusion Middleware Control](#)
- [Searching Log Files Using WLST](#)

11.3.2.1 Searching Log Files Using Fusion Middleware Control

You can search for diagnostic messages using standard and supplemental ODL attributes using the Log Messages page of Fusion Middleware Control. By default, this page shows a summary of the logged issues for the last hour.

You can modify the search criteria to identify messages of relevance. You can view the search results in different modes, allowing ease of navigation through large amounts of data.

The following sections describe how to search log files:

- [Searching Log Files: Basic Searches](#)
- [Searching Log Files: Advanced Searches](#)

11.3.2.1.1 Searching Log Files: Basic Searches This section describes how to perform basic searches for log messages.

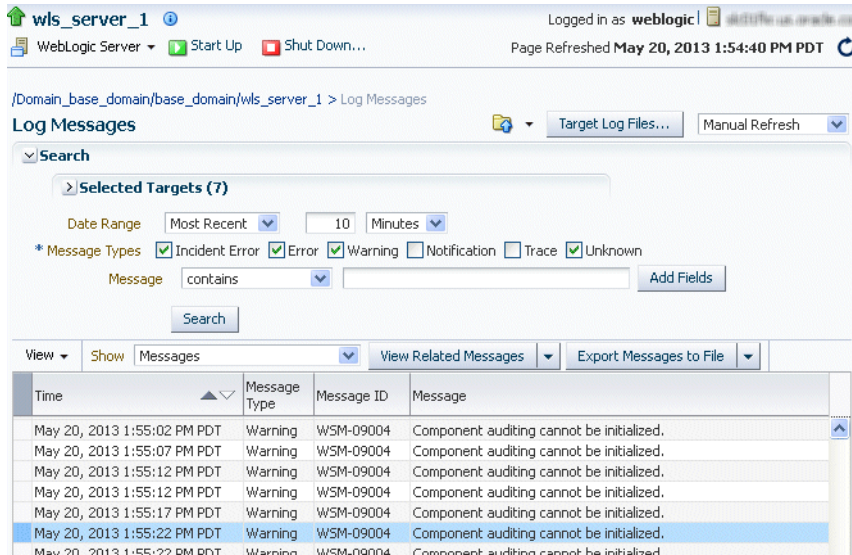
You can search for all of the messages for all of the entities in a domain, an Oracle WebLogic Server, a component, or an application.

For example, to search for messages for a domain:

1. From the WebLogic Domain menu, choose **Logs**, then **View Log Messages**.

To search for messages for a component or application, select the component or application. Then choose **Logs**, then **View Log Messages** from that target's menu.

The Log Messages page displays a Search section and a table that shows a summary of the messages for the last hour, as shown in the following figure:



2. In the Date Range section, you can select either:
 - **Most Recent:** If you select this option, select a time, such as 3 hours. The default is 1 hour.
 - **Time Interval:** If you select this option, select the calendar icon for **Start Date**. Select a date and time. Then, select the calendar icon for **End Date**. Select a date and time.
3. In the Message Types section, select one or more of the message types. The types are described in [Table 11-3](#).
4. You can specify more search criteria, as described in [Section 11.3.2.1.2](#).
5. Click **Search**.
6. To help identify messages of relevance, in the table, for **Show**, select one of the following modes:
 - **Messages:** Shows the matching messages.
To see the details of a particular message, click the message. The details are displayed below the table of messages.
To view related messages, select a message, then click **View Related Messages** and select **by Time** or **by ECID (Execution Context ID)**.
 - **Group by:** You can choose to group by one of the selections. This option groups the messages based on the criteria you selected, such as Message Type or ECID.
To see the messages, click the count in one of the message type columns. The Messages by Message Type page is displayed. To see the details of a particular message, click the message. The details are displayed below the table of messages.

11.3.2.1.2 Searching Log Files: Advanced Searches You can refine your search criteria using the following controls in the Log Messages page:

- **Message:** You can select an operator, such as **contains** and then enter a value to be matched.

- **Add Fields:** Click this to specify additional criteria, such as Host, which lets you narrow the search to particular hosts. Then click **Add**.
For each field you add, select an operator, such as **contains** and then enter a value to be matched.
- **Broaden Target Scope:** Click this to expand the search to logs associated with all members of the parent of the target. For example, if you are searching an application's logs, you can expand the search to contain the Managed Server to which the application is deployed.
- **Selected Targets:** Expand this to see the targets that are participating in the search. To add targets, click **Add** and provide information in the dialog box. To remove targets, select the target and click **Remove**.

11.3.2.2 Searching Log Files Using WLST

You can search the log files using the WLST `displayLogs` command. You can narrow your search by specifying criteria, such as time, component ID, message type, or ECID. For example:

- To search for error messages generated in the last 5 minutes, for a system component such as the Oracle HTTP Server `ohs1`, use the following command:

```
displayLogs(target='sc:ohs1',#last=5)
```

- To search for error messages generated in the last 10 minutes for the Managed Server `wls_server_1`, use the following command:

```
displayLogs(oracleInstance='/scratch/Oracle/config/domains/WLS_domain',#target='wls_server_1',#last=10)
```

You can narrow your search by using the `query` parameter and specifying criteria, such as component ID, message type, or ECID. In the `query` clause, you can specify a query expression with any of the attributes listed in [Table 11-1](#). Some of the criteria you can use are:

- Types of messages. For example, to search for `ERROR` and `INCIDENT_ERROR` messages for the Managed Server `wls_server_1`, use the following command:

```
displayLogs(oracleInstance='/scratch/Oracle/config/domains/wls_domain',######target='wls_server_1',#####query='MSG_TYPE#eq#ERROR#or#MSG_TYPE#eq#INCIDENT_ERROR')
```

- A particular ECID. For example, to search for error messages with a particular ECID (`0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0`) for the Managed Server `wls_server_1`, use the following command:

```
displayLogs(oracleInstance='/scratch/Oracle/config/domains/wls_domain',######target='wls_server_1',#####query='ecid#eq#0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0')
```

- Component type. For example, to search for messages from Oracle HTTP Server instances, use the following query:

```
displayLogs(query='COMPONENT_ID#eq#ohs')
```

- Range of time. To search for error messages that occurred within a specified range of time, you specify the attribute `TSTZ_ORIGINATING` with both `from` and `to` operators, using the following format:

```
displayLogs(query='TSTZ_ORIGINATING#from#start_time#and######TSTZ_ORIGINATING#to#end_time')
```

You specify the date using the following ISO 8601 time format:

```
YYYY-MM-DDThh:mm:ss-hh:mm_offset_from_UTC
```

For example:

```
2013-04-30T12:00:00:0000-08:00
```

For example, to display the error message from between 8:00 a.m. and 11 a.m. on April 17, 2013, use the following command:

```
displayLogs (query='TSTZ_ORIGINATING#from#2013-04-17T08:00:00-07:00#  
#####and#TSTZ_ORIGINATING#to#2013-04-17T11:00:00-07:00')
```

- **Group messages.** To display a count of messages, grouped by specific attributes, use the `groupBy` parameter to the WLST command `displayLogs`. For example, to display the count of WARNING messages by component, use the following command:

```
displayLogs (groupBy=[ 'COMPONENT_ID' ], #query='MSG_TYPE#eq#WARNING')
```

11.3.3 Downloading Log Files

You can download messages using Fusion Middleware Control or WLST commands, as described in the following topics:

- [Downloading Log Files Using Fusion Middleware Control](#)
- [Downloading Log Files Using WLST](#)

11.3.3.1 Downloading Log Files Using Fusion Middleware Control

You can download the log messages to a file. You can download either the matching messages from a search or the messages in a particular log file.

To download the matching messages from a search to a file using Fusion Middleware Control:

1. From the navigation pane, expand the domain and select the target, for example by clicking on the domain.
2. From the dynamic target menu, such as the WebLogic Domain menu, choose **Logs**, then **View Log Messages**.

The Log Messages page is displayed.

3. Search for particular types of messages as described in [Section 11.3.2.1](#).
4. Select a file type by clicking **Export Messages to File** and select one of the following:
 - **As Oracle Diagnostic Log Text (.txt)**
 - **As Oracle Diagnostic Log Text (.xml)**
 - **As Comma-Separated List (.csv)**

An Opening dialog box is displayed.

5. Select either **Open With** or **Save to Disk**. Click **OK**.

To export specific types of messages or messages with a particular Message ID to a file:

1. From the navigation pane, expand the domain. Select a Managed Server.

2. From the dynamic target menu, choose **Logs**, then **View Log Messages**.

The Log Messages page is displayed.

3. Search for particular types of messages as described in [Section 11.3.2.1](#).
4. For **Show**, select **Group by Message Type** or **Group by Message ID**.
5. To download the messages into a file, if you selected Group by Message Type, select the link in one of the columns that lists the number of messages, such as the Errors column. If you selected Group by Message ID, select one of the links in the Occurrences column.

The Messages by Message Type page or Message by Message ID is displayed.

6. Select a file type by clicking the arrow near **Export All to File**.

You can select one of the following:

- **As Oracle Diagnostic Log Text (.txt)**
- **As Oracle Diagnostic Log Text (.xml)**
- **As Comma-Separated List (.csv)**

An Opening dialog box is displayed.

7. Select either **Open With** or **Save to Disk**. Click **OK**.

To download the log files for a specific component using Fusion Middleware Control:

1. For system components, from the navigation pane, expand the installation type, such as **Web Tier** and select the component. For Java components, from the navigation pane, expand the component type, and then select the component.
2. From the dynamic target menu, choose **Logs**, then **View Log Messages**.

The Log Messages page is displayed.

3. Click **Target Log Files**.

The Log Files page is displayed. On this page, you can see a list of log files related to the component or application.

4. Select a log file and click **Download**.
5. An Opening dialog box is displayed.
6. Select either **Open With** or **Save to Disk**. Click **OK**.

11.3.3.2 Downloading Log Files Using WLST

You can download log files using the WLST `displayLogs` command and redirecting the output to a file. For example:

```
displayLogs(type=['ERROR', 'INCIDENT_ERROR'], #exportFile='/scratch/tmp/download_log.txt')
```

The messages are written to the file `download_log.txt` in the specified directory. By default, they are written to standard output.

11.4 Configuring Settings for Log Files

You can change the log settings of Managed Servers and Java components using Fusion Middleware Control or WLST.

Note: For many system components, which are listed in [Section 2.4.3](#), you cannot configure settings for log files using Fusion Middleware Control. For information about how to configure options for log files for system components, see the administrator's guide for the component.

For Java components, you can configure the names and locations of log files, the size of the log files, the level of information written to the log files, the format, and the Locale encoding, as described in the following topics:

- [Changing Log File Locations](#)
- [Configuring Log File Rotation](#)
- [Setting the Level of Information Written to Log Files](#)
- [Specifying the Log File Format](#)
- [Specifying the Log File Locale](#)

Note the following about using the WLST commands to configure log settings:

- To use the custom WLST logging commands, you must invoke the WLST script from the Oracle Common home. See [Section 2.4.2](#) for more information.
- The configuration commands, such as `setLogLevel`, only work in connected mode. That is, you must connect to a running WebLogic Server instance before you invoke the commands.

The configuration commands are supported for Java components that run within a WebLogic Server, but are not supported for Oracle WebLogic Server. The configuration commands are not supported for system components.

- Most of the WLST logging commands require that you are running in the `domainRuntime` tree. For example, to connect and to run in the `domainRuntime` tree, use the following commands:

```
connect('username',#'password',#'localhost:port_number')
domainRuntime()
```

- The `listLoggers`, `getLogLevel`, and `setLogLevel` commands work in `config` and `runtime` mode. In `config` mode the commands work on loggers that are defined in the configuration file. In `runtime` mode, the commands work directly with loggers that are defined in the server JVM. By default, the `setLogLevel` command sets the level on the run-time logger and updates the logger definition in the configuration file. By default, the `listLoggers` and `getLogLevel` commands return run-time loggers.

For more information about the commands, see "Logging Custom WLST Commands" in the *WLST Command Reference for Infrastructure Components*.

11.4.1 Changing Log File Locations

You can change the name and location of log files by using Fusion Middleware Control or WLST commands, as described in the following topics:

- [Changing Log File Locations Using Fusion Middleware Control](#)
- [Changing Log File Locations Using WLST](#)

11.4.1.1 Changing Log File Locations Using Fusion Middleware Control

To change the name and location of a component log file using Fusion Middleware Control:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs**, then **Log Configuration**.

The Log Configuration page is displayed.

Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration**, then **Log Configuration**.

3. Select the Log Files tab.
4. In the table, select the log handler and click **Edit Configuration**.

The Edit Log File dialog box is displayed, as shown in the following figure:

5. For **Log Path**, enter a new path.
6. Click **OK**.
7. In the confirmation window, click **Close**.

Note that if you change the location of Oracle HTTP Server log files, the location of the access_log and ohsn.log files are changed, but the location of console~OHS~1.log is not changed.

11.4.1.2 Changing Log File Locations Using WLST

To change the log file location using WLST, use the `configureLogHandler` command. For example, to change the path of the logger named `odl-handler`, use the following command:

```
configureLogHandler (name='odl-handler', #path='/scratch/Oracle/logs')
```

11.4.2 Configuring Log File Rotation

An **ODL log** is a set of log files that includes the current ODL log file and zero or more **ODL Archives (segment files)** that contain older messages. As the log file grows, new information is added to the end of the log file, `server_name-diagnostic.log`. When the log file reaches the rotation point, it is renamed and a new log file, `server_name-diagnostic.log` is created. You specify the rotation point, by specifying the maximum ODL segment size or the rotation time and rotation frequency.

Segment files are created when the ODL log file `server_name-diagnostic.log` reaches the rotation point. That is, the `server_name-diagnostic.log` is renamed to `server_name-diagnostic-n.log`, where *n* is an integer, and a new `server_name-diagnostic.log` file is created when the component generates new diagnostic messages.

To limit the size of the ODL log, you can specify:

- The maximum size of the logging directory. Whenever the sum of the sizes of all of the files in the directory reaches the maximum, the oldest archive is deleted to keep the total size under the specified limit.

By default, the log files are rotated when they reach 10 MB. The maximum size of all log files for a particular component is 100 MB.

- The maximum size of the log file. You specify that a new log file be created when a specific time or frequency is reached.

Note: After you change the log file rotation, the configuration is reloaded dynamically. It may take 1 or 2 seconds to reload the configuration.

The following topics describe how to change the rotation:

- [Specifying Log File Rotation Using Fusion Middleware Control](#)
- [Specifying Log File Rotation Using WLST](#)

11.4.2.1 Specifying Log File Rotation Using Fusion Middleware Control

To configure log file rotation using Fusion Middleware Control:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs**, then **Log Configuration**.

The Log Configuration page is displayed.

Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration**, then **Log Configuration**.

3. Select the Log Files tab.
 4. In the table, select the logger and click **Edit Configuration**.
- The Edit Log File dialog box is displayed.
5. In the Rotation Policy section, you can select one of the following:

- **Size Based:** If you select this, enter the following:
 - For **Maximum Log File Size**, enter the size in MB, for example, 15.
 - For **Maximum Size of All Log Files**, enter the size in MB, for example, 150.
- **Time Based:** If you select this, enter the following:
 - For **Start Time**, click the calendar and select the date and time when you want the rotation to start. For example, select September 8, 2010 6:00 AM.
 - For **Frequency**, you can select **Minutes** and enter the number of minutes, or you can select **Hourly**, **Daily**, or **Weekly**.

- For **Retention Period**, you can specify how long the log files are kept. You can select **Minutes** and enter the number of minutes, or you can specify **Day, Week, Month, or Year**.

Specifying a shorter period means that you use less disk space, but are not able to retrieve older information.

6. Click **OK**.

7. In the confirmation window, click **Close**.

11.4.2.2 Specifying Log File Rotation Using WLST

To specify log file rotation using WLST, use the `configureLogHandler` command. You can specify size-based rotation or time-based rotation.

For example, to specify that the log files rotate daily and that they are retained for a week, use the following command:

```
configureLogHandler (name='odl-handler', #rotationFrequency='daily',
#####retentionPeriod='week')
```

To specify that the size of a log file does not exceed 5 MB and rotates when it reaches that size, use the following command:

```
configureLogHandler (name='odl-handler', #maxFileSize='5M')
```

11.4.3 Setting the Level of Information Written to Log Files

You can configure the amount and type of information written to log files by specifying the message type and level. For each message type, possible values for the message level are from 1 (lowest severity) through 32 (highest severity). Some components support only some of the levels for each message type. See the administrator's guide for your component for more information. Generally, you need to specify only the type; you do not need to specify the level.

When you specify the type, Oracle Fusion Middleware returns all messages of that type, as well as the messages that have a higher severity. For example, if you set the message type to `WARNING`, Oracle Fusion Middleware also returns messages of type `INCIDENT_ERROR` and `ERROR`.

[Table 11–3](#) describes the message types and the most common levels for each type.

Table 11–3 Diagnostic Message Types and Level

Message Type	Level	Description
INCIDENT_ERROR	1	A serious problem that may be caused by a bug in the product and that should be reported to Oracle Support. Examples are errors from which you cannot recover or serious problems.
ERROR	1	A serious problem that requires immediate attention from the administrator and is not caused by a bug in the product. An example is if Oracle Fusion Middleware cannot process a log file, but you can correct the problem by fixing the permissions on the document.
WARNING	1	A potential problem that should be reviewed by the administrator. Examples are invalid parameter values or a specified file does not exist.

Table 11–3 (Cont.) Diagnostic Message Types and Level

Message Type	Level	Description
NOTIFICATION	1	A major lifecycle event such as the activation or deactivation of a primary sub-component or feature. This is the default level for NOTIFICATION.
NOTIFICATION	16	A finer level of granularity for reporting normal events.
TRACE	1	Trace or debug information for events that are meaningful to administrators, such as public API entry or exit points.
TRACE	16	Detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.
TRACE	32	Very detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.

The default is NOTIFICATION, level 1.

The INCIDENT_ERROR, ERROR, WARNING, and NOTIFICATION with level 1 have no performance impact. For other types and levels, note the following:

- NOTIFICATION, with level 16: Minimal performance impact.
- TRACE, with level 1: Small performance impact. You can enable this level occasionally on a production environment to debug problems.
- TRACE, with level 16: High performance impact. This level should not be enabled on a production environment, except on special situations to debug problems.
- TRACE, with level 32: Very high performance impact. This level should not be enabled in a production environment. It is intended to be used to debug the product on a test or development environment.

Table 11–4 shows the log level mappings among ODL format, Oracle WebLogic Server, and Java.

Table 11–4 Mapping of Log Levels Among ODL, Oracle WebLogic Server, and Java

ODL	WebLogic Server	Java
OFF	OFF	2147483647 - OFF
INCIDENT_ERROR:1	(EMERGENCY)	1100
INCIDENT_ERROR:4	EMERGENCY	1090
INCIDENT_ERROR:14	ALERT	1060
INCIDENT_ERROR:24	CRITICAL	1030
ERROR:1	(ERROR)	1000 - SEVERE
ERROR:7	ERROR	980
WARNING:1	WARNING	900 - WARNING
WARNING:7	NOTICE	880
NOTIFICATION:1	INFO	800 - INFO
NOTIFICATION:16	(DEBUG)	700 - CONFIG
TRACE:1	(DEBUG)	500 - FINE
TRACE:1	DEBUG	495

Table 11–4 (Cont.) Mapping of Log Levels Among ODL, Oracle WebLogic Server, and

ODL	WebLogic Server	Java
TRACE:16	(TRACE)	400 - FINER
TRACE:32	(TRACE)	300 - FINEST
TRACE:32	TRACE	295

You can configure the message levels using Fusion Middleware Control or WLST commands, as described in the following topics:

- [Configuring Message Levels Using Fusion Middleware Control](#)
- [Configuring Message Levels Using WLST](#)

11.4.3.1 Configuring Message Levels Using Fusion Middleware Control

You can set the message level for a particular log file or for loggers.

To set the message level for a component log file:

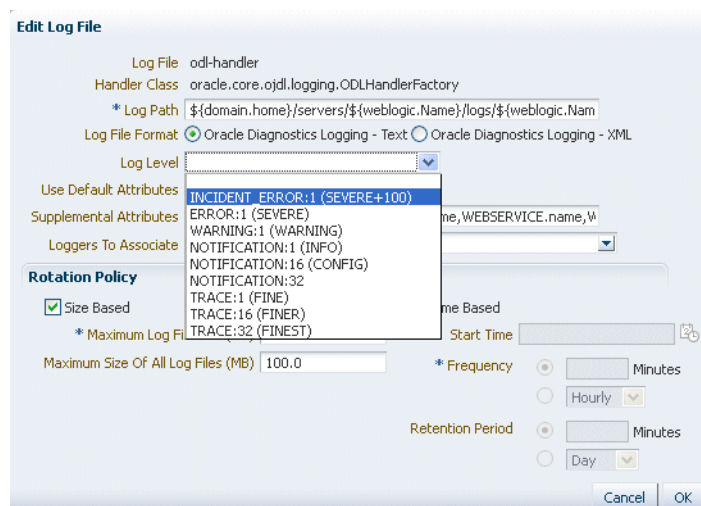
1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs**, then **Log Configuration**.

The Log Configuration page is displayed.

Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration**, then **Log Configuration**.

3. Select the Log Files tab.
4. In the table, select the log file and click **Edit Configuration**.

The Edit Log File dialog box is displayed, as shown in the following figure:



5. For **Log Level**, select the logging level. For example, select **WARNING:1 (WARNING)**.
6. Click **OK**.
7. In the confirmation window, click **Close**.

To set the message level for one or more loggers for a component:

1. From the navigation pane, select the component.

2. From the dynamic target menu, choose **Logs**, then **Log Configuration**.
The Log Configuration page is displayed.
Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration**, then **Log Configuration**.
3. Select the **Log Levels** tab.
4. For **View**, select **Runtime Loggers** or **Loggers with Persistent Log Level State**.
Run-time loggers are loggers that are currently active. Persistent loggers are loggers that are saved in a configuration file and the log levels of these loggers are persistent across component restarts. A run-time logger can also be a persistent logger, but not all run-time loggers are persistent loggers.
5. In the table, to specify the same level for all loggers, select the logging level for the top-level logger. Then, for child loggers that do not specify that the logging level is inherited from the parent, specify **Inherited from Parent**. For most situations, that is sufficient.

However, if you need to specify the level for a particular logger, expand the logger and then, for the logger that you want to modify, select the logging level. For example, for the logger oracle.wsm.management.logging, select **WARNING:1 (WARNING)**.
6. Click **Apply**.

11.4.3.2 Configuring Message Levels Using WLST

To set the message level with WLST, you use the `setLogLevel` command. To get the current message level, you use the `getLogLevel` command. You must be connected to WebLogic Server before you use the configuration commands.

You can view the log level for a logger for an Oracle WebLogic Server. For example, to view the log level of the Oracle WebLogic Server `wls_server_1`, use the following command:

```
getLogLevel(logger='oracle',##target='wls_server_1')
```

```
NOTIFICATION:1
```

You can set the log level for a particular logger. The following example sets the message type to `WARNING` for the logger `oracle.wsm.msg.logging`:

```
setLogLevel(target='wls_server_1',#logger='oracle.wsm.msg.logging',#level='WARNING')
```

To get a list of loggers for the Oracle WebLogic Server `wls_server_1`, use the `listLoggers` command:

```
listLoggers(target='wls_server_1')
.
.
.
oracle.wsm.msg.logging#####|#NOTIFICATION:1
oracle.wsm.nobehavior.model.NoBehaviorAssertion#####|#<Inherited>
oracle.wsm.policy.advertisement.AdvertisementContext####|#<Inherited>
oracle.wsm.policy.model.impl.AndCompositeAssertion#####|#<Inherited>
.
.
.
```

You can also filter logger names using the pattern parameter and a regular expression. For example, to return all loggers that begin with `oracle` in the Oracle WebLogic Server `wls_server_1`, use the following command:

```
listLoggers(target='wls_server_1',#pattern='oracle.*')
```

```
-----
Logger#####|Level#####
-----
oracle#####NOTIFICATION:1
#oracle.adf#####<Inherited>
oracle.adf.controller#####<Inherited>
oracle.adf.desktopintegration#####<Inherited>
oracle.adf.faces#####<Inherited>
```

11.4.4 Specifying the Log File Format

By default, information is written to log files in ODL text format. You can change the format to ODL XML format using Fusion Middleware Control or WLST commands, as described in the following topics:

- [Specifying the Log File Format Using Fusion Middleware Control](#)
- [Specifying the Log File Format Using WLST](#)

11.4.4.1 Specifying the Log File Format Using Fusion Middleware Control

To change the format using Fusion Middleware Control:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs**, then **Log Configuration**.

The Log Configuration page is displayed.

Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration**, then **Log Configuration**.

3. Select the Log Files tab.
 4. In the table, select the log file and click **Edit Configuration**.
- The Edit Log File dialog box is displayed.
5. For Log File Format, select **Oracle Diagnostics Logging - Text** or **Oracle Diagnostics Logging - XML**.

6. Click **OK**.

7. In the confirmation window, click **Close**.

11.4.4.2 Specifying the Log File Format Using WLST

To specify the log file format using WLST, you use the `configureLogHandler` command, with the `format` parameter and specify either ODL-Text or ODL-XML. ODL-Text is the default.

For example, to specify ODL-XML format, use the following command:

```
configureLogHandler(name='odl-handler',#format='ODL-XML')
```

11.4.5 Specifying the Log File Locale

The language and data formats used in the log files are determined by the default locale of the server Java Virtual Machine (JVM). You can change them using the

Language and Regional Options applet in Control Panel on Windows or the LANG and LC_ALL environment variables on a UNIX platform.

The character encoding of log files is determined by the server JVM's default character encoding or an optional configuration setting. You should choose an encoding that supports all languages used by the users, or the log file may be corrupted. By default, the log is in the server JVM's default character encoding. If you change the encoding, delete or rename old log files to prevent them from being damaged by the new logs appended in a different encoding.

For support of any language, Oracle recommends that you use Unicode UTF-8 encoding. On a UNIX operating system, setting the LANG and LC_All environment variables to a locale with the UTF-8 character set enables UTF-8 logging (for example, en_US.UTF-8 for the US locale in UTF-8 encoding).

You can specify the log file locale using WLST commands or by editing a file, as described in the following topics:

- [Specifying the Log File Encoding Using WLST](#)
- [Specifying the Log File Encoding in logging.xml](#)

11.4.5.1 Specifying the Log File Encoding Using WLST

To specify the log file encoding using WLST, use the `configureLogHandler` command. You can use the encoding parameter to specify the character set encoding.

For example, to specify UTF-8, use the following command:

```
configureLogHandler (name="odl-handler", #encoding="UTF-8")
```

11.4.5.2 Specifying the Log File Encoding in logging.xml

To specify the log file encoding in the logging.xml file, use an optional encoding property to specify the character set encoding.

The logging.xml file is located in the following directory:

```
DOMAIN_HOME/config/fmwconfig/servers/server_name/
```

For example, to specify UTF-8, add the following encoding property in the log_handler element:

```
<property#name='encoding' #value='UTF-8' />
```

11.5 Correlating Messages Across Log Files and Components

Oracle Fusion Middleware components provide **message correlation** information for diagnostic messages. Message correlation information helps those viewing diagnostic messages to determine relationships between messages across components. Each diagnostic message contains an **Execution Context ID (ECID)** and a **Relationship ID (RID)**:

- An ECID is a globally unique identifier associated with the execution of a particular request. An ECID is generated when the request is first processed.
- A RID distinguishes the work done in one thread on one process, from work done by any other threads on this and other processes on behalf of the same request.

The ECID and RID help you to use log file entries to correlate messages from one application or across Oracle Fusion Middleware components. By searching for related messages using the message correlation information, multiple messages can be examined and the component that first generates a problem can be identified (this

technique is called **first-fault component isolation**). Message correlation data can help establish a clear path for a diagnostic message across components, within which errors and related behavior can be understood.

You can use the ECID and RID to track requests as they move through Oracle Fusion Middleware.

The following shows an example of an ECID:

```
0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0
```

The RID is one or more numbers separated by a colon (:). The first RID created for a request is 0. Each time work is passed from a thread that has an ECID associated with it to another thread or process, a new RID is generated that encodes the relationship to its creator. That is, a new generation is created. Each shift in generation is represented by a colon and another number. For example, the seventh child of the third child of the creator of the request is:

```
0:3:7
```

You can view all the messages with the same ECID using the WLST `displayLogs` command. The following example searches for the ECID in the domain:

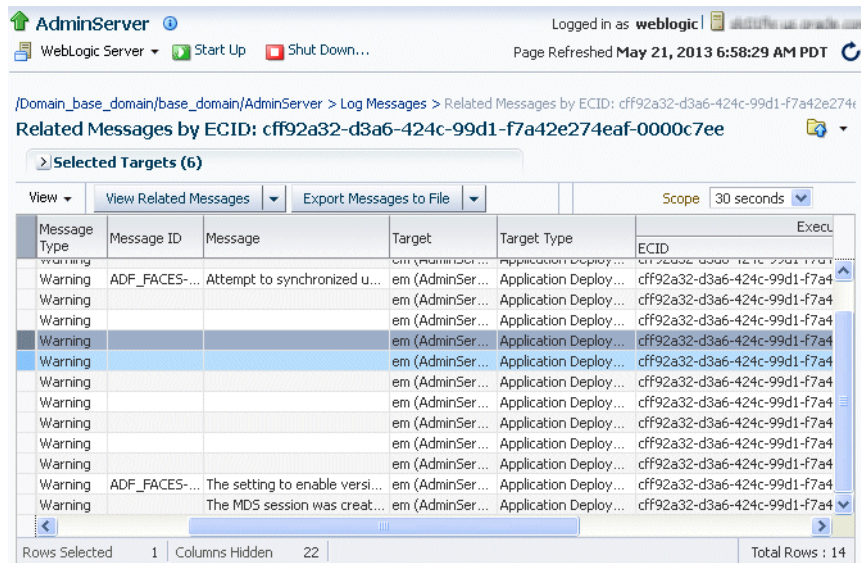
```
displayLogs(ecid='0000H19TwKUCs1T6uBi8UH181kWX000002')
```

You can also search for the ECID in a WebLogic Server instance, or a system component, by specifying it in the target option.

You can search for messages with a particular ECID on the Log Messages page in Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Logs**, then **View Log Messages**.
To search for messages for a component or application, select the component or application and then choose **Logs**, then **View Log Messages** from that target's menu.
2. Specify search criteria, as described in [Section 11.3.2.1.2](#).
3. Click **Search**.
4. Select a message, then click **View Related Messages** and select **by ECID (Execution Context ID)**.

The messages with the same ECID are displayed, as shown in the following figure:



- Trace the ECID to the earliest message. (You may need to increase the scope to view the first message with the ECID.)

11.6 Configuring Tracing

Sometimes you need more information to troubleshoot a problem than it is usually recorded in the logs. One way to achieve that is to increase the level of messages logged by one or more components. For example, you can set the logging level to TRACE:1 or TRACE:32, as described in [Section 11.4.3](#), which results in more detailed messages being written to the log files. This is referred to as **tracing**.

However, this can often result in a large amount of log messages being written to the log files. Oracle Fusion Middleware provides the following mechanisms to fine-tune which messages are traced:

- QuickTrace, which provides fine-grained logging to memory
- Selective Trace, which provides fine-grained logging for a specific user or other properties of a request

The following topics provide information about how to use these mechanisms:

- [Configuring and Using QuickTrace](#)
- [Configuring and Using Selective Tracing](#)

11.6.1 Configuring and Using QuickTrace

With QuickTrace, you can trace messages from specific loggers and store the messages in memory. Because QuickTrace logs the messages to memory, it avoids the cost of formatting, string manipulations, and input/output operations. As a result, you can enable fine-level application logging for specific loggers without performance overhead.

By default, QuickTrace writes the messages to one common buffer. However, you can specify that messages for particular users are written to separate buffers.

You can save the messages that are in memory to a file by invoking the QuickTrace Dump in Fusion Middleware Control as described in [Section 11.6.1.1.2](#) or by using the WLST, as described in [Section 11.6.1.2.2](#).

To enable QuickTrace, you create a QuickTrace handler and associate a logger with it. You can specify the buffer size, as well as other attributes, for the handler. Then, you set the level of the amount and type of information to be written by the loggers to memory.

The following topics describe how to enable and use QuickTrace:

- [Configuring QuickTrace Using Fusion Middleware Control](#)
- [Configuring QuickTrace Using WLST](#)

11.6.1.1 Configuring QuickTrace Using Fusion Middleware Control

You can configure and use QuickTrace using Fusion Middleware Control, as described in the following topics:

- [Configuring QuickTrace Using Fusion Middleware Control](#)
- [Writing the Trace Messages to a File Using Fusion Middleware Control](#)

11.6.1.1.1 Configuring QuickTrace Using Fusion Middleware Control To configure QuickTrace using Fusion Middleware Control:

1. From the navigation pane, expand the domain. Right-click the Managed Server name and choose **Logs**, then **Log Configuration**.

The Log Configuration page is displayed.

2. Select the QuickTrace tab.
3. Click **Create**.

The Create QuickTrace Handler dialog box is displayed, as shown in the following figure:

4. For **Name**, enter a name for the handler.
5. For **Buffer Size**, enter the size, in bytes, for the buffer for storing log messages in memory. The default is 5242880.
6. For **Maximum Field Length**, enter the length, in bytes, for each field in a message. The fields can include the message text, supplemental attributes, and the thread name. The default is 240.

An excessively long field for each message can reduce the amount of log records in the buffer.

7. For **Handler Level**, select the log level for the handler. See [Section 11.4.3](#) for information about the levels.
8. For **Loggers to Associate**, select the loggers that you want to associate with this QuickTrace handler. All messages of the specified level for these handlers will be written to memory.

Many loggers are associated with other handlers. For example, the oracle.adf logger is associated with the handlers odl-handler, wls-domain, and console-handler. When you set the level of the logger, these handlers will use the same level, such as TRACE:1, for the logger, such as oracle.adf. As a result, much information will be written to the log files, consuming resources. To avoid consuming resources, set the level of the handlers to a lower level, such as WARNING or INFORMATION.

9. Select **Enable User Buffer?** if you want to enable a user buffer. If you enable this, the handler maintains an individual buffer for each user you specify.

Then, for **User Names for Reserve Buffer**, enter the names of the users, separated by commas.

10. For the remaining options, accept the default values. For information about the options, see "ConfigureLogHandler" in the *WLST Command Reference for Infrastructure Components*.

11. Click **OK**.

12. When the configuration completes processing, click **OK**.

Now, messages of the specified level for the specified loggers are written to memory.

11.6.1.1.2 Writing the Trace Messages to a File Using Fusion Middleware Control You can save the messages that are in memory to a file by invoking the QuickTrace Dump in Fusion Middleware Control:

1. From the QuickTrace tab of the Log Configuration page, select the handler and click **Invoke QuickTrace Dump**.

The Invoke QuickTrace Dump dialog box is displayed.

2. For **Buffer Name**, if you have specified user buffers when you configured the QuickTrace handler, select the user, or select Common Buffer for users that you did not specify. If you did not specify any user buffers, the Common Buffer is the only option.

3. Click **OK**.

When the processing is complete, the View Log Messages page is displayed.

4. You can search the messages, as described in [Section 11.3.2](#), and you can correlate the messages as described in [Section 11.5](#).

In addition, you can download the messages to a file, as described in [Section 11.3.3.1](#).

11.6.1.2 Configuring QuickTrace Using WLST

You can configure and use QuickTrace Using WLST, as described in the following topics:

- [Configuring QuickTrace Using WLST](#)

- [Writing the Trace Messages to a File Using WLST](#)
- [Disabling QuickTrace Using WLST](#)

11.6.1.2.1 Configuring QuickTrace Using WLST To configure QuickTrace using WLST, you associate a logger with the QuickTrace handler, using the `configureLogHandler` command.

For example, to associate the `oracle.adf` logger with the QuickTrace handler and write all `TRACE:1` messages to memory:

1. Use the `configureLogHandler` command to associate the logger with the QuickTrace handler:

```
configureLogHandler (name="quicktrace-handler", #addToLogger="oracle.adf")

Handler#Name:#quicktrace-handler
type:###oracle.core.ojdl.logging.QuickTraceHandlerFactory
encoding:#####UTF-8
maxLength:#240
mode:###objRef
useThreadName:##false
useSourceClassandMethod:#####false
useLoggingContext:#####false
bufferSize:#####5242880
```

The messages for the handler are written to a common buffer.

You can set additional properties for the QuickTrace handler. For example, to enable user buffers for the users `user1` and `user2`:

```
configureLogHandler (name="quicktrace-handler", #addToLogger="oracle.adf.faces",
#####propertyName="enableUserBuffer", #propertyValue="true",
#####propertyName="enableUserBuffer", #propertyValue="user1, #user2")
...
Handler#Name:#quicktrace-handler
type:###oracle.core.ojdl.logging.QuickTraceHandlerFactory
useLoggingContext:#####false
bufferSize:#####5242880
#.
#.
#.
reserveBufferUserID:###user1, #user2
enableUserBuffer:#####true
```

Messages for `user1` and `user2` are written to separate buffers. In addition, messages related to other users are written to the common buffer.

To confirm the settings for the handler, use the `listLogHandlers` command, as described in "listLogHandlers" in the *WLST Command Reference for Infrastructure Components*.

2. Set the level of the logger, using the `setLogLevel` command:

```
setLogLevel (logger='oracle.adf', #level='TRACE:1')
```

To confirm the settings for the logger, use the `listLoggers` command, as described in "listLoggers" in the *WLST Command Reference for Infrastructure Components*.

3. Many loggers are associated with other handlers. For example, the `oracle.adf` logger is associated with the handlers `odl-handler`, `wls-domain`, and `console-handler`. When you set the level of the logger, these handlers will use the same level (`TRACE:1`) for the logger `oracle.adf`. As a result, much information will

be written to the log files, consuming resources. To avoid consuming resources, set the level of the handlers to a lower level, such as WARNING or INFORMATION.

For this example, set the level of the three handlers to WARNING:1:

```
configureLogHandler (name="odl-handler", #level="WARNING:1")
configureLogHandler (name="wls-domain", #level="WARNING:1")
configureLogHandler (name="console-handler", #level="WARNING:1")
```

Note that you should keep the level of the QuickTrace handler at ALL, which is the default.

For more information, see "configureLogHandler" in the *WLST Command Reference for Infrastructure Components*

To confirm the level for the handler, use the getLogLevel command, as described in [Section 11.4.3.2](#).

11.6.1.2.2 Writing the Trace Messages to a File Using WLST You can save the messages to a file by using the executeDump command.

For example:

```
executeDump (name="odl.quicktrace", #outputFile="/scratch/oracle1/qt1.dmp")
```

The command writes the dump to the specified file.

For more information about the executeDump command, see [Section 12.4.4.3](#).

In addition, if an incident is created (automatically or manually), the QuickTrace messages are written to dump files in the incident directory. If you enabled user buffers, each user will have one file and the common buffer will have one file.

The file names have the following format:

```
odl_quicktraceN_iincident_number.username.dmp
```

For example:

```
odl_quicktrace6_i1.weblogic.dmp
```

See [Section 12.4.6.1](#) for information about creating an incident.

11.6.1.2.3 Disabling QuickTrace Using WLST To disable QuickTrace, use the configureLogHandler command and specify that the level is OFF:

```
configureLogHandler (name="quicktrace-handler", #level="OFF")
```

```
Handler#Name:#quicktrace-handler
type:###oracle.core.ojdl.logging.QuicktraceHandlerFactory
#.
#.
#.
reserveBufferUserID:###user1,#user2
enableUserBuffer:#####true
```

To remove a specific logger from association with the QuickTrace handler, use the configureLogHandler command with the removeFromLogger parameter:

```
configureLogHandler (name="quicktrace-handler", #
removeFromLogger="oracle.adf.faces")

Handler#Name:#quicktrace-handler
type:###oracle.core.ojdl.logging.QuicktraceHandlerFactory
```

```
reserveBufferUserID:####user1,#user2
enableUserBuffer:#####true
```

For more information, see "configureLogHandler" in the *WLST Command Reference for Infrastructure Components*.

11.6.2 Configuring and Using Selective Tracing

Selective tracing provides fine-grained logging for specified users or other attributes of a request.

For example, a user cannot perform some functions because of security permissions, but it is not clear what operations or lack of permission for those operations are posing a problem.

In this case, you can enable tracing across the entire system but this would generate a large volume of log messages for all users in the system, not only for the user having a problem. With selective tracing, you can enable tracing only for the user who is having a problem. Then, you can ask the user to retry the functions. Following that, you can look at the trace messages which apply to the specific request made by the user.

You can also specify the logger to narrow the scope of the messages being logged.

You can use Selective Tracing using Fusion Middleware Control or WLST, as described in the following topics:

- [Configuring Selective Tracing Using Fusion Middleware Control](#)
- [Configuring Selective Tracing Using WLST](#)

11.6.2.1 Configuring Selective Tracing Using Fusion Middleware Control

You can configure selective tracing, view traces, and disable selective tracing using Fusion Middleware Control, as described in the following topics:

- [Configuring Selective Tracing Using Fusion Middleware Control](#)
- [Viewing Selective Traces Using Fusion Middleware Control](#)
- [Disabling Selective Tracing Using Fusion Middleware Control](#)

11.6.2.1.1 Configuring Selective Tracing Using Fusion Middleware Control To configure selective tracing using Fusion Middleware Control:

1. From the navigation pane, right-click the domain name and choose **Logs**, then **Selective Tracing**.

The Selective Tracing page is displayed, as shown in the following figure:

The screenshot shows the 'Selective Tracing' configuration page. At the top, it indicates the user is logged in as 'weblogi' and the page was refreshed on May 22, 2013 at 11:08:15 AM PDT. The page title is 'base_domain' and 'WebLogic Domain'. The main heading is 'Selective Tracing' with a sub-heading 'Use this page to configure the selective tracing settings.' An 'Information' section states that selective tracing should be used for critical diagnostics and adds diagnostic logging messages. The 'Tracing Options' section includes a search bar for 'Application Name', an 'Add Fields' button, a 'Level' dropdown set to 'TRACE:32 (FINEST)', a 'Description' text box, and a 'Duration (minutes)' field set to '30'. Under 'Trace ID', the 'Generate A New Unique Trace ID' radio button is selected. Below are sections for 'ODL' (ODL logging events, Enable checked) and 'DMS' (DMS data is composed of metrics, Enable unchecked). At the bottom, there is a 'Loggers' section with a right-pointing arrow.

2. For **Application Name**, select an application.
3. To add more fields, click **Add Fields** and select one of the options, such as Client Host or User Name.
4. For **Level**, select a logging level. [Table 11-3](#) describes the logging levels.
5. For **Description**, enter a description.
6. For **Duration**, enter the number of minutes that you want the selective trace to run.

The selective trace is disabled after the specified time.

7. For Trace ID, select either **Generate a New Unique Trace ID** or **Use a Custom Trace ID**. If you select Use a Custom Trace ID, enter an ID of your choosing, but make sure that it is unique. Note Fusion Middleware Control does not verify the uniqueness of the ID.
8. In the ODL section, select **enable**.
9. In the Loggers section, by default, all loggers are selected.

You can select specific loggers that you want to trace. To find particular loggers, you can enter a string in the field above the table and click the Return key. For example, to find all loggers that begin with oracle.security, enter oracle.security.

Then, in the table, select the loggers in the **Enable on All Servers** column.

Note when you select loggers, those loggers apply to all current and active traces. Also note that even if you disable the loggers, you may see messages because all loggers have a general logging level, such as Notification. Those messages would still be written.

10. Click **Start Tracing**.

Now that you have started the trace, you can view active traces, as well as former traces, as described in [Section 11.6.2.1.2](#).

11.6.2.1.2 Viewing Selective Traces Using Fusion Middleware Control You can view the selective traces that are currently active and the history of selective traces.

To view the selective traces:

1. From the Selective Tracing page, select the **Active Traces and Tracing History** tab.

The tab shows a table with the active traces and a table with the tracing history, as shown in the following figure:

The screenshot shows the 'Selective Tracing' page with the 'Active Traces And Tracing History' tab selected. The page includes an information section, a 'Tracing Options' section, and two tables. The 'Active Traces' table has one row with the following data:

Trace ID	Option Name	Option Value	Description	Start Time
f7c7331a-0c24-4d63-9f6d-1ffd7da5bd16	Application Name	mdsappdb1		May 22, 2013 11:08:15 AM PDT

The 'Tracing History' table is empty, showing 'No tracing history'.

2. To view a trace, select it from the appropriate table.

The Log Messages page is displayed, with the messages that were captured by Selective Tracing. You can search the messages, as described in [Section 11.3.2](#), and you can correlate the messages as described in [Section 11.5](#).

In addition, you can download the messages to a file, as described in [Section 11.3.3.1](#).

11.6.2.1.3 Disabling Selective Tracing Using Fusion Middleware Control To disable selective tracing using Fusion Middleware Control:

1. From the navigation pane, right-click the domain name and choose **Logs**, then **Selective Tracing**.
2. Select the **Active Traces and Tracing History** tab.
3. In the Active Traces table, select the trace and click **Disable**.

11.6.2.2 Configuring Selective Tracing Using WLST

You can configure selective tracing, view traces, and disable selective tracing using WLST, as described in the following topics:

- [Configuring Selective Tracing Using WLST](#)

- [Viewing Selective Traces Using WLST](#)
- [Disabling Selective Traces Using WLST](#)

11.6.2.2.1 Configuring Selective Tracing Using WLST You can configure loggers for selective tracing and start tracing using the WLST `configureTracingLoggers` and `startTracing` commands.

For the simplest case, you can configure and start a trace using the `startTracing` command. When you do so, the selective tracing includes all loggers enabled for selective tracing.

For example, `user1` receives errors when attempting to perform certain operations. To start a trace of messages related to `user1` and to set the logging level to `FINE`, use the following command:

```
startTracing(user="user1", level="FINE")
Started#tracing#with#ID:#885649f7-8efd-4a7a-9898-accbfc0bbba3#
```

The `startTracing` command does not provide options to include or exclude particular loggers. In this case, you can use the `configureTracingLoggers` command. This command allows you to configure selective tracing to include only particular loggers and particular Oracle WebLogic Server instances. Note that the options you specify apply to all current and active traces.

For example, to configure selective tracing to include only security-related loggers:

1. Specify that all loggers be disabled for tracing, as shown in the following example:

```
configureTracingLoggers(action="disable")
Configured#1244#loggers
```

2. Enable the security-related loggers, by specifying the `pattern` option with a regular expression:

```
configureTracingLoggers(pattern='oracle.security.*', #action="enable")
Configured#62#loggers
```

To see a list of the loggers that support selective tracing, use the WLST `listTracingLoggers` command, as shown in the following example:

```
listTracingLoggers(pattern="oracle.security.*")
-----+-----
Logger#####|#Status#
-----+-----
oracle.security#####|#enabled
oracle.security.audit.logger#####|#enabled
oracle.security.jps.az.common.util.JpsLock#####|#enabled
#
#
#
```

3. Use the `startTracing` command, specifying the users and the level. For example:

```
startTracing(user="user1", level="FINE")
Started#tracing#with#ID:#a9580e65-13c4-420b-977e-5ba7dd88ca7f
```


See Also: The following commands in the *WLST Command Reference for Infrastructure Components* for complete syntax:

- "configureTracingLoggers"
- "startTracing"
- "listTracingLoggers"

11.6.2.2.2 Viewing Selective Traces Using WLST After you have begun a trace, you can see the active traces by using the `listActiveTraces` command, as shown in the following example:

```
listActiveTraces()
-----+-----+-----+-----+-----+-----
Trace#ID#####|Attr.#Name|Attr.#Value|#Level|#Exp.#Time#|Description
-----+-----+-----+-----+-----+-----
b73b351c-9a9b-47df-b05a-356a336d5780#|#USER_ID##|#user1#####|#FINE#|#5/22/13#11:17#AM#|#
a9580e65-13c4-420b-977e-5ba7dd88ca7f#|#USER_ID###|#user1#####|#FINE#|#5/22/13#11:19#AM#|#
```

You can view the contents of the trace using the `displayLogs` command and passing it the trace ID. You can also view traces that have stopped. For example:

```
displayLogs("a9580e65-13c4-420b-977e-5ba7dd88ca7f")
```

See "listActiveTraces" in the *WLST Command Reference for Infrastructure Components* for complete syntax.

11.6.2.2.3 Disabling Selective Traces Using WLST To avoid excessive logging in the system, you can disable a selective trace when you have obtained the information that you need. To disable a selective trace, you use the `WLST stopTracing` command, passing it the trace ID or user. For example:

```
stopTracing(traceId="885649f7-8efd-4a7a-9898-accbfc0bbba3")
Stopped#1#traces
```

You can also disable all traces by using the `stopAll` option. For example:

```
stopTracing(stopAll=1)
```

See "stopTracing" in the *WLST Command Reference for Infrastructure Components* for complete syntax

Diagnosing Problems

This chapter describes how to use the Oracle Fusion Middleware Diagnostic Framework to collect and manage information about a problem so that you can resolve it or send it to Oracle Support for resolution.

This chapter contains the following sections:

- [Understanding the Diagnostic Framework](#)
- [How the Diagnostic Framework Works](#)
- [Configuring the Diagnostic Framework](#)
- [Investigating, Reporting, and Solving a Problem](#)
- [Managing and Running the Health Test Framework](#)

12.1 Understanding the Diagnostic Framework

Oracle Fusion Middleware includes a Diagnostic Framework, which aids in detecting, diagnosing, and resolving problems. The problems that are targeted in particular are critical errors such as those caused by code bugs, metadata corruption, customer data corruption, deadlocked threads, and inconsistent state.

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error (such as log files) are immediately captured and tagged with this number. The data is then stored in the Automatic Diagnostic Repository (ADR), where it can later be retrieved by incident number and analyzed.

The goals of the Diagnostic Framework are:

- First-failure diagnosis
- Limiting damage and interruptions after a problem is detected
- Reducing problem diagnostic time
- Reducing problem resolution time
- Simplifying customer interaction with Oracle Support

The Diagnostic Framework includes the following technologies:

- **Automatic capture of diagnostic data upon first failure:** For critical errors, the ability to capture error information at first failure greatly increases the chance of a quick problem resolution and reduced downtime. The Diagnostic Framework automatically collects diagnostics, such as thread dumps, DMS metric dumps, and WebLogic Diagnostics Framework (WLDF) server image dumps. Such diagnostic data is similar to the data collected by airplane "black box" flight recorders. When a problem is detected, alerts are generated and the fault diagnosability

infrastructure is activated to capture and store diagnostic data. The data is stored in a file-based repository and is accessible with command-line utilities.

- **Standardized log formats:** Standardized log formats (using the ODL log file format) across all Oracle Fusion Middleware components allows administrators and Oracle Support personnel to use a single set of tools for problem analysis. Problems are more easily diagnosed, and downtime is reduced.
- **Diagnostic rules:** Each component defines diagnostic rules that are used to evaluate whether a given log message should result in an incident being created and which dumps should be executed. The diagnostic rules also indicate whether an individual dump should be created synchronously or asynchronously.

In addition, you can define custom rules that apply to a domain, a server, or an application in a domain or server.

- **Incident detection log filter:** The incident detection log filter implements the `java.util.logging` filter. It inspects each log message to see if an incident should be created, basing its decision on the diagnostic rules for components and applications.
- **Incident packaging service (IPS) and incident packages:** The IPS enables you to automatically and easily gather the diagnostic data—log files, dumps, reports, and more—pertaining to a critical error that has a corresponding incident, and package the data into a zip file for transmission to Oracle Support. All diagnostic data relating to a critical error that has been detected by the Diagnostics Framework is captured and stored as an incident in ADR. The incident packaging service identifies the required files automatically and adds them to the zip file.

Before creating the zip file, the IPS first collects diagnostic data into an intermediate logical structure called an incident **package**. Packages are stored in the Automatic Diagnostic Repository. If you choose to, you can access this intermediate logical structure, view and modify its contents, add or remove additional diagnostic data at any time, and when you are ready, create the zip file from the package and upload it to Oracle Support.

- **Integration with WebLogic Diagnostics Framework (WLDF):** The Oracle Fusion Middleware Diagnostics Framework integrates with some features of WebLogic Diagnostics Framework (WLDF), including the capturing of WebLogic Server images on detection of critical errors. WLDF is a monitoring and diagnostic framework that defines and implements a set of services that run within WebLogic Server processes and participate in the standard server life cycle. Using WLDF, you can create, collect, analyze, archive, and access diagnostic data generated by a running server and the applications deployed within its containers. This data provides insight into the run-time performance of servers and applications and enables you to isolate and diagnose faults when they occur.

Oracle Fusion Middleware Diagnostics Framework integrates with the following components of WLDF:

- WLDF Watch and Notification, which watches specific logs and metrics for specified conditions and sends a notification when a condition is met. There are several types of notifications, including JMX notification and a notification to create a Diagnostic Image. Oracle Fusion Middleware Diagnostics Framework integrates with the WLDF Watch and Notification component to create incidents.
- Diagnostic Image Capture, which gathers the most common sources of the key server state used in diagnosing problems. It packages that state into a single

artifact, the Diagnostic Image. With Oracle Fusion Middleware Diagnostics Framework, it writes the artifact to ADR.

For more information about WLDF, see *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

12.1.1 About Incidents and Problems

To facilitate diagnosis and resolution of critical errors, the Diagnostic Framework introduces two concepts for Oracle Fusion Middleware: problems and incidents.

A **problem** is a critical error. Critical errors manifest as internal errors or other severe errors. Problems are tracked in the ADR. Each problem has a **problem key**, which is a text string that describes the problem. It includes an error code (in the format *XXX-nnnnn*) and in some cases, other error-specific values.

An **incident** is a single occurrence of a problem. When a problem (critical error) occurs multiple times, an incident is created for each occurrence. Incidents are timestamped and tracked in the ADR. Each incident is identified by a numeric incident ID, which is unique within the ADR home. When an incident occurs, the Diagnostic Framework:

- Gathers first-failure diagnostic data about the incident in the form of dump files (incident dumps).
- Stores the incident dumps in an ADR subdirectory created for that incident.
- Registers the incidents dumps with the incident in ADR.

12.1.1.1 Incident Flood Control

It is conceivable that a problem could generate dozens or perhaps hundreds of incidents in a short period of time. This would generate too much diagnostic data, which would consume too much space in the ADR and could possibly slow down your efforts to diagnose and resolve the problem. For these reasons, the Diagnostic Framework applies flood control to incident generation after certain thresholds are reached. A **flood-controlled incident** is an incident that is not recorded in the ADR. Instead, the Diagnostic Framework writes a message at the WARNING level to the log file and returns an `oracle.dfw.incident.Incident` object. Flood-controlled incidents provide a way of informing you that a critical error is ongoing, without overloading the system with diagnostic data.

By default, if more than 5 incidents with the same problem key occur within 60 minutes, subsequent incidents with the same problem key are flood controlled. You can change this value using MBeans, as described in [Section 12.3](#).

12.1.2 Diagnostic Framework Components

The following topics describe the key components of the Diagnostic Framework:

- [Automatic Diagnostic Repository](#)
- [Diagnostic Dumps](#)
- [Management MBeans](#)
- [WLST Commands for Diagnostic Framework](#)
- [ADRCI Command-Line Utility](#)

Note: To use the Diagnostic Framework, in particular the Automatic Diagnostic Repository, the Managed Servers must have Oracle JRF applied. The following directory will exist for each Managed Server if Oracle JRF has been applied:

DOMAIN_HOME/SERVERS/server_name/adr

If the directory does not exist take one of the following steps:

- Apply Oracle JRF, as described in [Section 18.3.1](#).
 - If Oracle JRF has been applied, restart the servers, making sure that the Node Manager property `startScriptEnabled` is set to `true`, as described in [Section 2.8.1](#).
-
-

12.1.2.1 Automatic Diagnostic Repository

The Automatic Diagnostic Repository (ADR) is a file-based hierarchical repository for Oracle Fusion Middleware diagnostic data, such as traces and dumps. The Oracle Fusion Middleware components store all incident data in the ADR. Each Oracle WebLogic Server stores diagnostic data in subdirectories of its own home directory within the ADR. For example, each Managed Server and Administration Server has an ADR home directory.

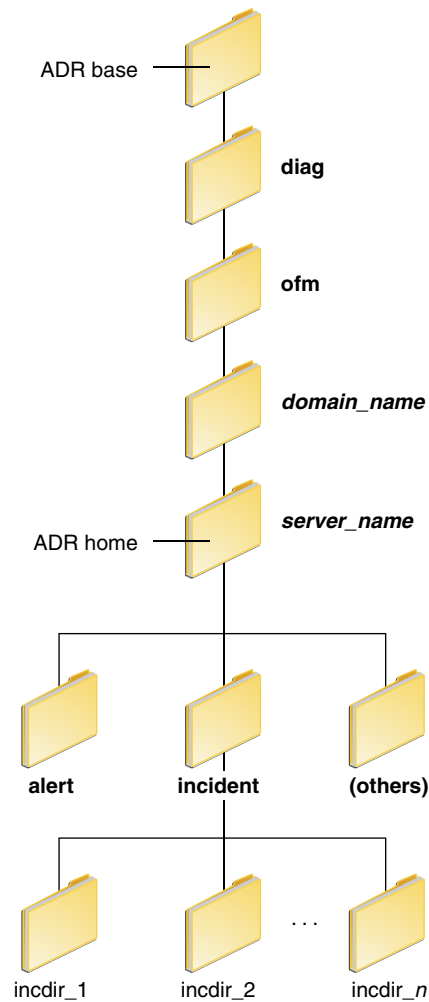
The ADR root directory is known as ADR base. By default, the ADR base is located in the following directory:

DOMAIN_HOME/servers/server_name/adr

Within ADR base, there can be multiple ADR homes, where each ADR home is the root directory for all incident data for a particular instance of Oracle WebLogic Server. The following path shows the location of the ADR home:

ADR_BASE/diag/ofm/domain_name/server_name

[Figure 12–1](#) illustrates the directory hierarchy of the ADR home for an Oracle WebLogic Server instance.

Figure 12–1 ADR Directory Structure for Oracle Fusion Middleware

The subdirectories in the ADR home contain the following information:

- alert: The XML-formatted alert log.
- incident: A directory that can contain multiple subdirectories, where each subdirectory is named for a particular incident. The subdirectories are named `incdir_n`, with n representing the number of the incident. Each subdirectory contains information and diagnostic dumps pertaining only to that incident.
- (others): Other subdirectories of ADR home, which store incident packages and other information.

Note: ADR uses the domain name as the Product ID and the server name as the Instance ID when it packages an incident. However, if either name is more than 30 characters, ADR truncates the name. In addition, dollar sign (\$) and space characters are replaced with underscores.

12.1.2.2 Diagnostic Dumps

A **diagnostic dump** captures and dumps specific diagnostic information when an incident is created (automatic) or on the request of an administrator (manual). When executed as part of incident creation, the dump is included with the set of incident diagnostics data. Examples of diagnostic dumps include a JVM thread dump, JVM class histogram dump, and DMS metric dump. For a list of diagnostic dumps, see [Table 12-7](#).

12.1.2.3 Management MBeans

The Diagnostic Framework provides MBeans that you can use to configure the Diagnostic Framework. For example, you can enable or disable flood control and you can configure how many incidents with the same problem key can occur within a specified time period. For information about using the management MBeans to configure the Diagnostic Framework, see [Section 12.3](#).

You can also use the MBeans to query and create incidents, discover the list of available diagnostic dump types, and execute individual diagnostic dumps.

12.1.2.4 WLST Commands for Diagnostic Framework

The Diagnostic Framework provides WLST commands that you can use to view information about problems and incidents, create incidents, execute specific dumps and query the set of diagnostic dump types. For more information, see:

- [Section 12.4.2.1, "Viewing Problems"](#)
- [Section 12.4.2.2, "Viewing Incidents"](#)
- [Section 12.4.4.1, "Listing Diagnostic Dumps"](#)
- [Section 12.4.4.2, "Viewing a Description of a Diagnostic Dump"](#)
- [Section 12.4.4.3, "Executing Dumps"](#)
- [Section 12.4.6.1, "Creating an Incident Manually"](#)
- "Diagnostic Framework Custom WLST Commands" in the *WLST Command Reference for Infrastructure Components*

To use the custom WLST Diagnostic Framework commands, you must invoke the WLST script from the Oracle Common home. See [Section 2.4.2](#) for more information.

12.1.2.5 ADRCI Command-Line Utility

The ADR Command Interpreter (ADRCI) is a utility that enables you to investigate problems, and package and upload first-failure diagnostic data to Oracle Support, all within a command-line environment. ADRCI also enables you to view the names of the dump files in the ADR, and to view the alert log with XML tags stripped, with and without content filtering.

ADRCI is installed in the following directory:

```
(UNIX) #ORACLE_HOME/bin  
(Windows) #ORACLE_HOME\bin
```

See the following sections for information about using the ADRCI command-line utility:

- [Section 12.4.6.3](#) for information on packaging an incident.
- [Section 12.4.6.4](#) for information on purging incidents.

See Also:

- The chapter "ADRCI: ADR Command Interpreter" in *Oracle Database Utilities*
- The chapter "Managing Diagnostic Data" in the *Oracle Database Administrator's Guide*

12.2 How the Diagnostic Framework Works

The Diagnostic Framework is active in each server and provides automatic error detection through predefined configured rules. Oracle Fusion Middleware components and applications automatically benefit from this always-on checking.

Incidents are automatically detected in two ways:

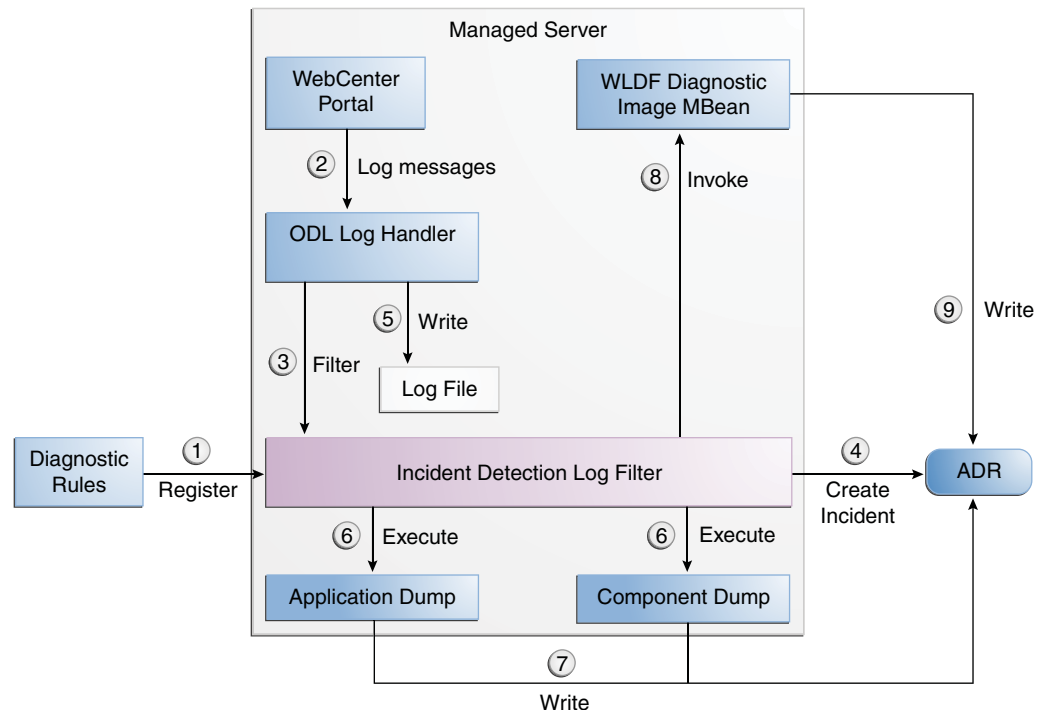
- By the incident detection log filter, which is automatically configured to detect critical errors.
- By the WLDF Watch and Notification component. The Diagnostics Framework listens for a predefined notification type and creates incidents when it receives such notifications.

For information about configuring WLDF Watch and Notification, see [Section 12.3.4](#).

- Programmatic incident creation. Some components create incidents directly.

[Figure 12-2](#) shows the interaction when the incident is detected by the incident log detector. It shows the interaction among the incident log detector, the WLDF Diagnostic Image MBean, ADR, and component or application dumps when an incident is detected by the incident log detector.

Figure 12-2 Incident Creation Generated by Incident Log Detector



The steps represented in [Figure 12-2](#) are:

1. The incident detection log filter is initialized with component and application diagnostic rules.
2. An application or component logs a message using the `java.util.logging` API.
3. The ODL log handler passes the message to the incident detection log filter.
4. The incident log detection filter inspects the log message to see if an incident should be created, basing its decision on the diagnostic rules for the component. If the diagnostic rule indicates that an incident should be created, it creates an incident in the ADR.
5. The ODL log handler writes the log message to the log file, and returns control to the application.

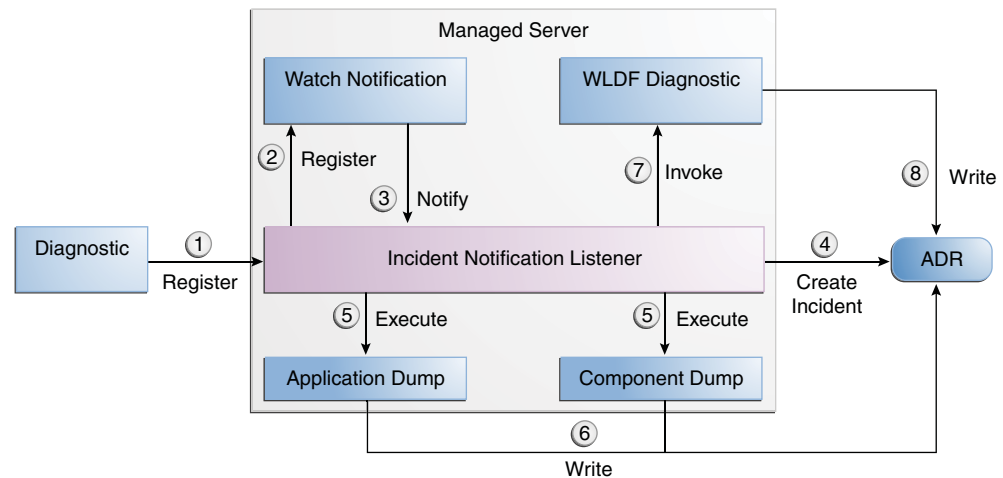
The ODL log handler writes the log message to the log file, and returns control to Oracle WebCenter Portal.

When an incident is created, a message, similar to the following, is written to the log file:

```
[2013-05-28T11:05:34.603-07:00]#[wls_server_1]#[NOTIFICATION]#[DFW-40101]#[oracle.dfw.incident]#[tid:#[ACTIVE].ExecuteThread: #'4' #for#queue:#'weblogic.kernel.Default#(self-tuning) ' ]#[userId:#weblogic]#[ecid:#66217af9-247f-4344-94a9-14f90e75a586-000e093f,0]#An#incident#has#been#signalled#with#the#incident#facts:#[problemKey=MDS-50500#[MANUAL]#incidentSource=MANUAL#incidentTime=Tue#May#28#11:05:34#PDT#2013#errorMessage=MDS-50500#executionContextId=null]
```

6. The Diagnostic Framework executes the diagnostic dumps that are indicated by the diagnostic rules for the component.
7. The Diagnostic Framework writes the dumps to ADR, in the directory created for the incident.
8. The Diagnostic Framework invokes the WLDF Diagnostic Image MBean requesting that a Diagnostic Image be created in ADR.
9. WLDF writes the Diagnostic Image to ADR.

Figure 12-3 shows the interaction when an incident is detected by the WLDF Watch and Notification system. It shows the interaction among the incident notification listener, the WLDF Watch and Notification system, and the WLDF Diagnostic Image MBean.

Figure 12–3 Incident Creation Generated by WLDF Watch Notification

The steps represented in [Figure 12–3](#) are:

1. The incident notification listener is initialized with component and application diagnostic rules.
2. Oracle Fusion Middleware Diagnostic Framework registers a JMX notification listener with WLDF. The listener listens for events from the WLDF Watch and Notification system. It only processes notifications of type `oracle.dfw.wldfnotification`.
3. Something in the system causes the configured WLDF watch to be triggered, causing a notification to be sent to the incident notification listener. The notification includes event information describing the data that caused the watch to trigger.
4. The Diagnostic Framework creates an incident in ADR.
5. The Diagnostic Framework executes the diagnostic dumps that are indicated by the diagnostic rules.
6. The Diagnostic Framework writes the dumps to ADR, in the directory created for the incident.
7. The Diagnostic Framework invokes the WLDF Diagnostic Image MBean requesting that a Diagnostic Image be created in ADR.
8. WLDF writes the Diagnostic Image to ADR.

12.3 Configuring the Diagnostic Framework

You can configure some settings for the Diagnostic Framework. In addition, you can configure an WLDF Watch and Notification to create an incident. The following topics describe how to configure the Diagnostic Framework:

- [Configuring Diagnostic Framework Settings](#)
- [Configuring Custom Diagnostic Rules](#)
- [Configuring Problem Suppression](#)
- [Configuring WLDF Watch and Notification for the Diagnostic Framework](#)

12.3.1 Configuring Diagnostic Framework Settings

You can configure the following settings:

- Enabling or disabling the detection of incidents through the log files
- Enabling or disabling flood control and setting parameters for flood control

You configure these settings by using the Diagnostic Framework MBean `DiagnosticConfig`. The following shows the MBean's `ObjectName`:

```
oracle.dfw:type=oracle.dfw.jmx.DiagnosticsConfigMBean,name=DiagnosticConfig#
```

[Table 12–1](#) shows the attributes for the `DiagnosticConfig` MBean and a description of each parameter.

Table 12–1 DiagnosticConfig MBean Attributes for Diagnostic Framework

Attributes	Description
<code>DumpSamplingIdleWhenHealthy</code>	Determines whether dump sampling is active when the system is healthy. By default, this is set to <code>true</code> , which means that dump sampling is not active until an incident occurs.
<code>DumpSamplingMinimumHealthyPeriod</code>	The amount of time in seconds that the dump sampling is active after an incident occurs. The default is 259200 seconds (72 hours).
<code>floodControlEnabled</code>	Enables or disables flood control. Specify <code>true</code> for enabled or <code>false</code> for disabled. The default is <code>true</code> . Note that flood control does not apply to manually created incidents.
<code>floodControlIncidentCount</code>	Sets the number of incidents with the same problem key that can be created within the time period, specified by <code>floodControlIncidentTimeoutPeriod</code> , before they are controlled by flood control. The default is 5. When flood control is enabled, if the number of incidents with the same problem key exceeds this count, no incidents are created, but the Diagnostic Framework writes a message at the <code>WARNING</code> level to the log file.
<code>floodControlIncidentTimeoutPeriod</code>	Sets the time period in which the number of incidents, as specified by <code>floodControlIncidentCount</code> , with the same problem key can be created before they are controlled by flood control. The default is 60 minutes.
<code>incidentCreationEnabled</code>	Enables or disables incident creation. Specify <code>true</code> for enabled or <code>false</code> for disabled. The default is <code>true</code> .
<code>logDetectionEnabled</code>	Enables or disables the detection of incidents through the log files. Specify <code>true</code> for enabled or <code>false</code> for disabled. The default is <code>true</code> .
<code>maxTotalIncidentSize</code>	Sets the maximum total size that is allocated for all incidents. When the limit is reached, the oldest incidents are purged until the space used by all incidents is less than the amount specified by this parameter. The default is 500 MB. The limit may be exceeded during the creation of an incident, but when the incident creation completes, the oldest incidents are purged.

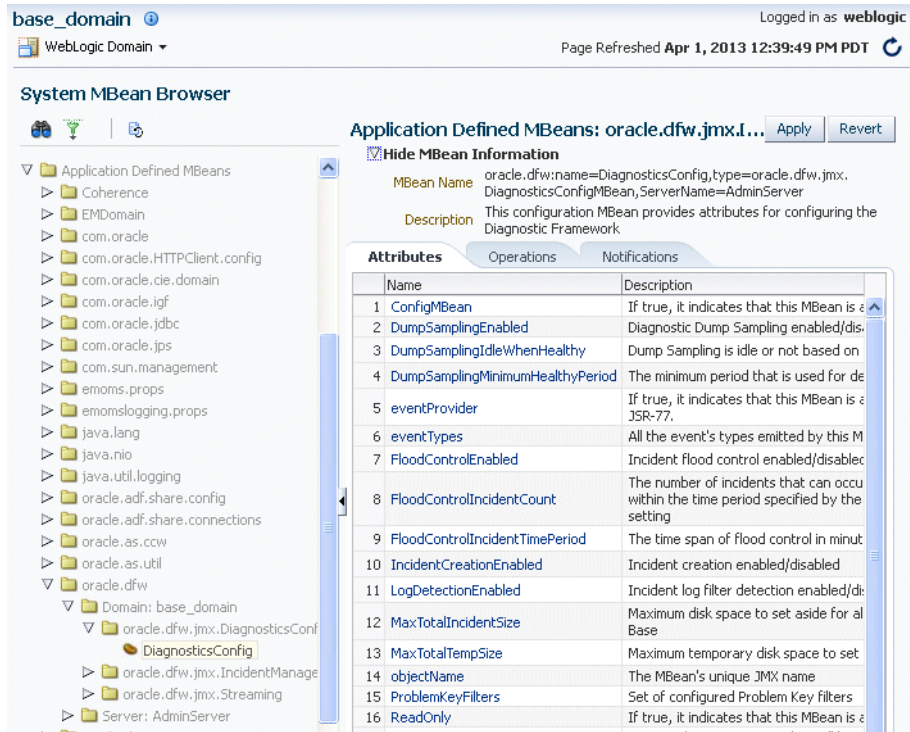
Table 12–1 (Cont.) DiagnosticConfig MBean Attributes for Diagnostic Framework

Attributes	Description
reservedMemoryKB	<p>The amount of reserved memory that is released when OutOfMemoryError is detected.</p> <p>When the Diagnostic Framework starts, it allocates 512 KB of memory for its own private use. When the Diagnostic Framework detects that an OutOfMemoryError has occurred in the server, it frees that block of memory and proceeds to create the incident.</p> <p>The default is 512 KB.</p>
uncaughtExceptionDetectionEnabled	<p>Enables the Java-based uncaught exception handler. When enabled and an uncaught exception is detected, an incident is created. Specify <code>true</code> for enabled or <code>false</code> for disabled.</p> <p>The default is <code>true</code>.</p>
useExternalCommands	<p>Indicates whether external JVM commands should be used to perform thread dumps. Specify <code>true</code> for enabled or <code>false</code> for disabled. The default is <code>true</code>.</p>

The following example shows how to configure these settings using the Fusion Middleware Control System MBean Browser:

1. From the WebLogic Domain menu, choose **System MBean Browser**.
The System MBean Browser page is displayed.
2. Expand **Application Defined Beans**, then `oracle.dfw`, then `domain.domain_name`, then `dfw.jmx.DiagnosticsConfigMBean`.
3. Select one of the **DiagnosticConfig** entries. There is one DiagnosticConfig entry for each server.
4. In the Application Defined MBean pane, expand **Show MBean Information** to see the server name.

The following shows the System MBean Browser page:



5. To change the values for the attributes listed in Table 12-1, enter or select the value in the Value field.
6. Click **Apply**.

12.3.2 Configuring Custom Diagnostic Rules

You can configure custom diagnostic rules that apply to a domain, a server, or an application in a domain or server.

You create the custom diagnostic rules by creating an .xml file with a particular format, which is shown in the example later in this section. You must save the file to one of the following locations:

- For rules that apply to the entire domain:


```
DOMAIN_HOME/config/fmwconfig/dfw
```
- For rules that apply to a particular server:


```
DOMAIN_HOME/config/fmwconfig/servers/server_name/dfw
```

The file name must use the following format:

```
name.xml
appname#name.xml
```

In the format, *appname* is the name of the application to which the rule applies. The *appname* must be the exact name of the deployed application. *name* is the name of the rule you specify. If you do not specify *appname*, the rules will apply to the entire server. For example, the following rule applies to the application myApp:

```
myApp#custom_rule.xml
```

The custom diagnostic rules file can contain the following types of elements to define the rule:

- Log detection conditions, which is optional

You can define a set of conditions, in the `logDetectionConditions` element, to check for in the diagnostic logs applicable to the server or to the specified application against which that the rules are registered. When a log message matching the condition is detected, an incident is created, capturing diagnostics that will help identify the problem. By default, all `INCIDENT_ERROR` messages are detected and an incident created for them. In addition, specific components may have configured rules to detect specific messages.

The following example shows a fragment of a custom diagnostic rules file that defines four log detection conditions. If one or more of the conditions are true, an incident is created.

```
<?xml#version="1.0"#encoding="UTF-8"?>
<diagnosticRules#xmlns="http://www.oracle.com/DFW/DiagnosticsFrameworkRules"#
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
##<logDetectionConditions>
#####<condition#messageSeverity="INCIDENT_ERROR"/>#
#####<condition#messageSeverity="ERROR"#component="jrfServer_admin"/>#
#####<condition#messageSeverity="ERROR"#module="test.servletA"/>#
#####<condition#messageId="FMW-40300"/>#
####</logDetectionConditions>
```

See [Table 12–2](#) for a description of the conditions you can use.

- Processing rules

You can define processing rules that are evaluated when either the server or application rules are involved in incident creation. For example, if the application `MyApp` is involved in incident creation, any rules associated with the `MyApp` application are evaluated. In all cases, server-wide rules are evaluated regardless of the application.

Processing rules consist of two parts:

- Default actions, which are optional. If they are present, they are always executed during incident creation. The actions are a list of diagnostic dumps to execute, along with optional arguments.

The following shows an example set of default actions:

```
<defaultActions>
#####<dumpAction#name="odl.logs">
#####<argument#name="timestamp"#value="INCIDENT_TIME"#valueType="fact"/>
#####</dumpAction>
#####<dumpAction#name="dms.metrics"/>
####</defaultActions>
```

See [Table 12–3](#) for a description of the optional arguments that you can use.

- Condition-based actions, which are executed only if the condition evaluates to true. Each `<rule>` element consists of a name attribute, along with a child `<ruleCondition>` element and a child `<ruleActions>` element. The `<ruleActions>` element contains one or more `dumpAction` elements. See [Table 12–4](#) for a list of the `<ruleCondition>` element attributes.

If multiple `<condition>` elements are specified in a single `<rule>` element, the `dumpAction` is executed only if all conditions evaluate to true.

The following shows an example of a condition-based action rule. If the MESSAGE_ID is DFW-99997, the condition evaluates to true and the jvm.classhistogram dump is executed.

```
<processingRules>
#####<rule#name="OOM">
#####<ruleCondition>
#####<condition#name="MESSAGE_ID"#value="DFW-99997"/>
#####</ruleCondition>
#####<ruleActions>
#####<dumpAction#name="jvm.classhistogram"/>
#####</ruleActions>
#####</rule>
#####</processingRules>
```

Table 12–2 describes the attributes you can use to create the log detection conditions:

Table 12–2 Conditions for the LogDetectionConditions Element

Condition	Description
messageSeverity	The log level at which the message was logged. (This is the MESSAGE_LEVEL field for ODL log files.) For example, INCIDENT_ERROR, ERROR.
messageId	The ID of the message. (This is the MESSAGE_ID field for ODL log files.) For example, DFW-99997.
component	The component name. (This is the COMPONENT_ID field for ODL log files.) For example, oracle.mds.
module	The name of the module that originated the message. (This is the MODULE_ID field for ODL log files.)

See Table 11–1 for a description of the ODL log file fields.

Table 12–3 describes the optional arguments that you can use for the <defaultActions> element.

Table 12–3 Optional arguments for the defaultActions Element

Argument	Description
name	The name of the argument.
value	The value of the argument
type	The type of argument. Valid values are: <ul style="list-style-type: none"> ▪ literal: If you specify this type, the literal value of the argument is used. This is the default. ▪ fact: If you specify this type, the value must be either INCIDENT_TIME or ECID. ▪ context: If you specify this type, the value must be the name of a value in the DMS Execution Context. For information on the DMS Execution Context, see "DMS Execution Context" in <i>Tuning Performance</i>.

Table 12–4 shows the <ruleCondition> element attributes.

Table 12–4 Attributes for the ruleCondition Element

Element	Description
name	The name of the attribute. Valid values depend on the valueType: <ul style="list-style-type: none"> ■ If the valueType is fact, valid values are COMPONENT_ID, MODULE_ID, or MESSAGE_ID. ■ If the valueType is context, the value must be the name of a value in the DMS Execution Context. For information on the DMS Execution Context, see "DMS Execution Context" in <i>Tuning Performance</i>.
operator	The operator. Value values are EQ, EQNoCase, NE, Contains, StartsWith, EndsWith, LT, GT, LE, GE. The default is EQ. The values are case sensitive.
value	The literal value to compare.
datatype	The data type. Valid values are String or Integer. The default is String. The values are case sensitive.
valueType	The type of argument: <ul style="list-style-type: none"> ■ fact ■ context

To create and load a custom diagnostic rule:

1. Create a file that contains the custom rules.

The following shows a sample custom rules file:

```
<?xml#version="1.0"#encoding="UTF-8"?>
<diagnosticRules#xmlns="http://www.oracle.com/DFW/DiagnosticsFrameworkRules"#
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
#
####
####<logDetectionConditions>
#####<condition#messageSeverity="INCIDENT_ERROR"/>#
#####//#detect#all#message#logged#at#level#INCIDENT_ERROR
#####<condition#messageSeverity="ERROR"#component="jrfServer_admin"/>#
#####//#detect#all#"jrfServer_admin"#component#messages#logged#at#level#
ERROR
#####<condition#messageSeverity="ERROR"#module="test.servletA"/>#
#####//#detect#all#"test.servlet"#module#messages#logged#at#level#ERROR
#####<condition#messageId="FMW-40300"/>#
#####//#detect#message#"FMW-40300"
####</logDetectionConditions>
####
####<defaultActions>
#####<dumpAction#name="odl.logs">
#####<argument#name="timestamp"#value="INCIDENT_TIME"#valueType="fact"/>
#####</dumpAction>
#####<dumpAction#name="dms.metrics"/>
####</defaultActions>
#
####<processingRules>
#####<rule#name="OOME">
#####<ruleCondition>
#####<condition#name="MESSAGE_ID"#value="DFW-99997"/>
#####</ruleCondition>
```

```
#####<ruleActions>
#####<dumpAction#name="jvm.classshistogram" />
#####</ruleActions>
#####</rule>
#####</processingRules>
#
</diagnosticRules>
```

2. Save the file, naming it with the extension `.xml`. If the rule applies to an application, precede the file name with `app_name#`. Save the file to one of the following locations:

```
DOMAIN_HOME/config/fmwconfig/dfw
DOMAIN_HOME/config/fmwconfig/servers/server_name/dfw
```

3. Load the rules, using the WLST command `reloadCustomRules`. The following example loads the rule `customrules.xml`, which applies to the application `myApp`:

```
reloadCustomRules (name='myApp#customrules.xml')
```

You can reload all the rules in the domain or all the rules that pertain to a particular server. The following example reloads all the rules for the server `wls_server1`:

```
reloadCustomRules (server='wls_server1')
```

For more information about the `reloadCustomRules` command, see "reloadCustomRules" in the *WLST Command Reference for Infrastructure Components*.

12.3.3 Configuring Problem Suppression

In certain situations, you may want to suppress the creation of incidents based on a particular problem key. For example, in a development environment, when you are developing a servlet, you may generate high number of uncaught exceptions as you refine the code. This results in the creation of unnecessary incidents.

The Diagnostic Framework allows you to configure problem suppression filters so that problems that match the filter criteria do not result in the creation of an incident.

When you configure a problem suppression filter, you use a regular expression that represents a pattern that you want to match. The regular expression is matched using the `java.util.regex` class. For example:

- The following regular expression matches any incident with a problem key that starts with `MDS-5000`.
`MDS-5000.*`
- The following regular expression matches any problem with the text `OutOfMemory`. Because the regular expression is case-sensitive, it will not match problems with the text `outofmemory`.
`.*OutOfMemory.*`

You can add and remove filters and get a list of filters or the detail of one filter using the `DiagnosticConfig` MBean.

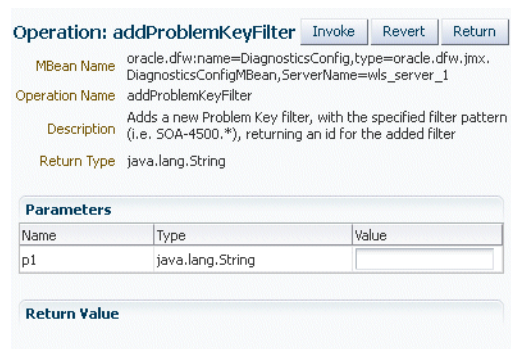
[Table 12-5](#) shows the operations and attribute for the configuring problem suppression filters and a description of each.

Table 12–5 DiagnosticConfig MBean Operations and Attributes for Problem Suppression Filters

Operations and Attribute	Description
Operation: addProblemKeyFilter(<i>filter_pattern</i>)	Adds a new problem suppression filter. You pass it the regular expression that represents a pattern that you want to match. For example: addProblemKeyFilter(".*OutOfMemory.*")
Attribute: getProblemKeyFilters()	Returns a list of the configured problem suppression filters. For example: getProblemKeyFilters()
Operation: getProblemKeyFilter(<i>filterID</i>)	Returns the filter pattern associated with the specified ID. For example: getProblemKeyFilter(<i>id</i>) To find the ID, use the getProblemKeyFilters() operation.
Operation: removeProblemKeyFilter(<i>filterID</i>)	Removes the filter pattern associated with the given filter ID. For example: removeProblemKeyFilter(<i>id</i>)

To configure a problem suppression filter:

1. From the WebLogic Domain menu, choose **System MBean Browser**.
The System MBean Browser page is displayed.
2. Expand **Application Defined Beans**, then **oracle.dfw**, then **domain.domain_name**, then **dfw.jmx.DiagnosticsConfigMBean**.
3. Select one of the **DiagnosticConfig** entries. There is one DiagnosticConfig entry for each server.
4. In the Application Defined MBeans pane, select the Operations tab.
5. Click **addProblemKeyFilter**. The Operation: addProblemKeyFilter page is displayed, as shown in the following figure:



6. For **Value**, enter a regular expression that represents a pattern that you want to match pattern. For example, in a development environment, you might want to add a filter so that incidents are not created when java.lang.IllegalStateException Java Exceptions are reported. In that case, enter the following:

```
".* [java.lang.IllegalStateException] .*"
```

7. Click **Invoke**.
8. Click **Return** to return to the Application Defined MBeans page.

You can delete the filters using the `removeProblemKeyFilter` operation.

You can retrieve a specific filter, passing the ID of the filter to the `getProblemKeyFilter` operation.

Alternatively, you can retrieve a list of the filters using the `getProblemKeyFilters` attribute:

1. From the WebLogic Domain menu, choose **System MBean Browser**.
The System MBean Browser page is displayed.
2. Expand **Application Defined Beans**, then `oracle.dfw`, then `domain.domain_name`, then `dfw.jmx.DiagnosticsConfigMBean`.
3. Select one of the **DiagnosticConfig** entries. There is one `DiagnosticConfig` entry for each server.
4. In the Application Defined MBeans pane, select the Attributes tab.
5. Click **ProblemKeyFilters**.
The list of problem suppression filters is displayed.

12.3.4 Configuring WLDF Watch and Notification for the Diagnostic Framework

Oracle Fusion Middleware configures a WLDF Diagnostics Module that contains a set of Watch and Notification rules for detecting a specific set of critical errors and creating an incident for each occurrence of those errors. The module is called `Module-FMWDFW` and contains the following set of Watch conditions:

Name	Description
Deadlock	Two or more Java threads have circular lock chains among their Java Monitor object usage.
StuckThread	An Oracle WebLogic Server <code>ExecuteThread</code> , which is blocked or busy for more than the time specified by the Oracle WebLogic Server <code>StuckThreadMaxTime</code> parameter.
UncheckedException	This category includes all <code>UncheckedException</code> , <code>RuntimeException</code> , and Errors caught by the Oracle WebLogic Server <code>ExecuteThread</code> , such as <code>NullPointerException</code> , <code>StackOverflowError</code> , or <code>OutOfMemoryError</code> .

The Diagnostic Module also includes a configured WLDF JMX Notification `FMWDFW-notification` of type `oracle.dfw.wldfnotification`. You can reuse this WLDF JMX Notification for your own WLDF Watch conditions to create an incident:

1. Display the Administration Console, as described in [Section 2.3.1](#).
2. In the left pane, expand **Diagnostics** and select **Diagnostic Modules**.
The Summary of Diagnostic Modules page is displayed.
3. Click **Module-FMWDFW**.
The Settings for Module-FMWDFW page is displayed.
4. Select the Watches and Notifications tab, which is shown in the following figure:

Settings for Module-FMWDFW

Configuration | Control | Targets

General | Collected Metrics | **Watches and Notifications** | Instrumentation

Save

Use this page to create and configure watches and notifications for this diagnostic module.

Enabled Specifies whether the Watch Notification component is enabled. [More Info...](#)

Severity: Notice The default notification severity level for all watches. When a watch triggers, the severity level is delivered with the notification. [More Info...](#)

Log Watch Severity: Warning The threshold severity level of log messages evaluated by log watch rules. Messages with a lower severity than this value will be ignored and not evaluated against the watch rules. [More Info...](#)

Save

Watches | Notifications

Use this page to add watches to the current diagnostic module and to configure those watches. Click the name of an existing watch to configure that watch.

[Customize this table](#)

Watches

New Delete Showing 1 to 4 of 4 Previous Next

<input type="checkbox"/>	Name ↕	Type	Enabled	Alarm Type
<input type="checkbox"/>	Deadlock	Server Log	true	Automatic Reset
<input type="checkbox"/>	HeapSpace	Server Log	true	Automatic Reset

5. Select the **Watches** tab and click **New**.

The Create Watch page is displayed.

6. For **Watch Name**, enter a name for the watch.

You can enter any name. Alternatively, you can use the following format to force the Diagnostic Framework to use a custom message ID:

```
message-id#[application_name]#any_text
```

The message ID consists of a prefix that can be 1 to 6 characters, and a number, that can be 1 to 6 digits. The application name is optional. For example:

```
WLS-40500#My_Watch_Name
```

The following example uses the application name testapp:

```
WLS-40501#testapp#My_Watch_Name
```

The Diagnostic Framework uses the message ID as the incident message ID in constructing the incident problem key.

7. For **Watch Type**, select a type, for example, Server Log.
8. Click **Next**.
9. For **Current Watch Rule**, construct an expression. For example, to construct the expression (SEVERITY = 'Error') AND (MSGID = 'BEA-000337'):
 - a. Click **Add Expressions**.

- b. For **Message Attribute**, select Severity.
 - c. For **Operator**, select =.
 - d. For **Value**, enter ERROR.
 - e. Click **OK**.
 - f. Click **Add Expressions**.
 - g. For **Message Attribute**, select MSGID.
 - h. For **Operator**, select =.
 - i. For **Value**, enter BEA-000337.
 - j. Click **OK**.
 - k. In the Create Watch page, ensure that the operator selected is **AND**.
10. Click **Next**.
 11. Select an alarm type and click **NEXT**.
 12. For **Notifications**, select **FMWDFW-notification** and move it to the Chosen box.
 13. Click **Finish**.

For more information on creating watches, see "Construct watch rule expressions" in the *Oracle WebLogic Server Administration Console Online Help*.

12.4 Investigating, Reporting, and Solving a Problem

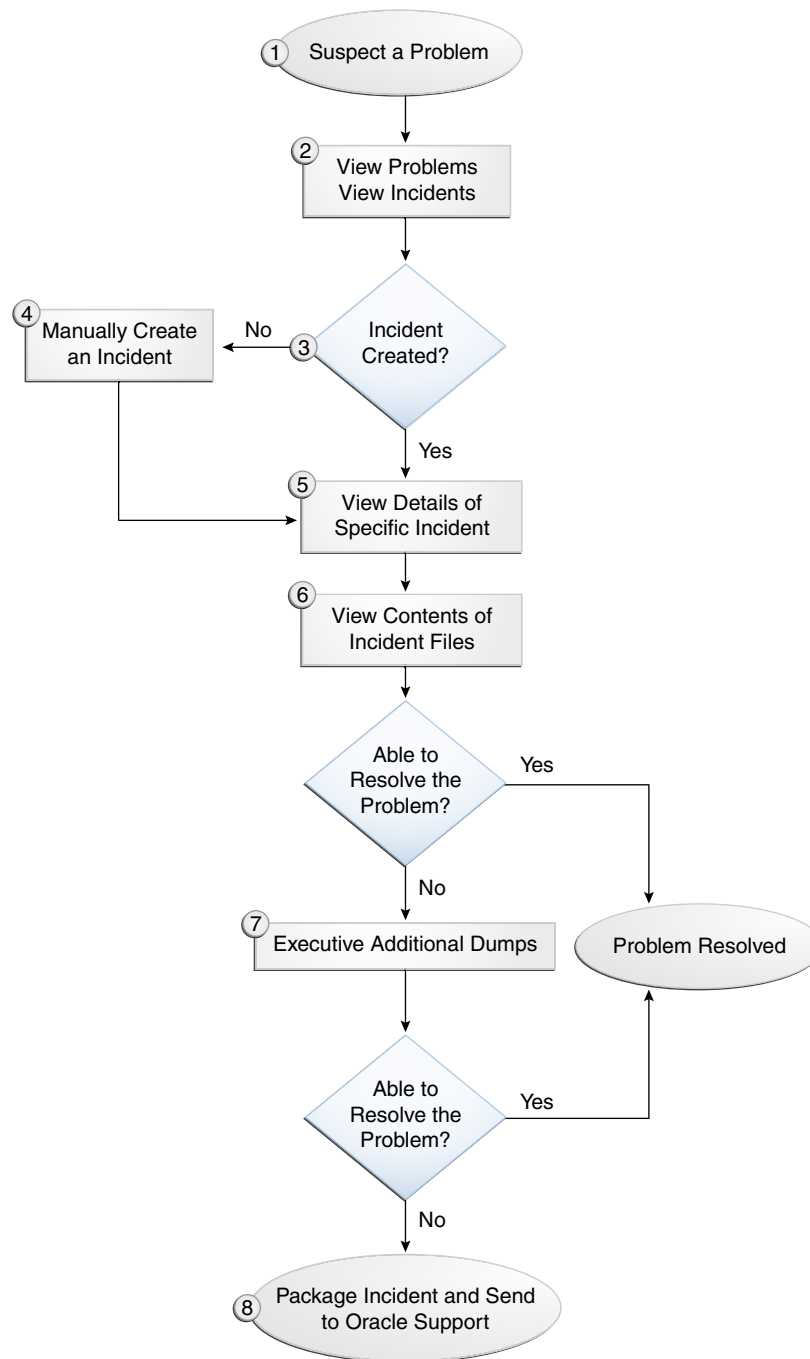
This section describes how to use WLST and ADRCI commands and Remote Diagnostic Agent (RDA) to investigate and report a problem (critical error), and in some cases, resolve the problem. The section begins with a roadmap that summarizes the typical set of tasks that you must perform. It describes the following topics:

- [Roadmap—Investigating, Reporting, and Resolving a Problem](#)
- [Viewing Problems and Incidents](#)
- [Analyzing Specific Problem Keys](#)
- [Working with Diagnostic Dumps](#)
- [Configuring and Using Diagnostic Dump Sampling](#)
- [Managing Incidents](#)
- [Generating an RDA Report](#)

12.4.1 Roadmap—Investigating, Reporting, and Resolving a Problem

Typically, investigating, reporting, and resolving a problem begins with a critical error. This section provides an overview of that workflow.

[Figure 12-4](#) illustrates the tasks that you complete to investigate, report, and resolve a problem.

Figure 12–4 Flow for Investigating a Problem

The following describes the workflow illustrated in [Figure 12–4](#):

1. You notice that the system, component, or application is not functioning as expected. For example, you notice that there is a performance problem or users have reported that the application that they are trying to access is reporting errors.
2. Check to see if a problem and an incident have been created that may be related to the symptoms you are observing:
 - a. View the set of problems by using the WLST `listProblems` command, as described in [Section 12.4.2.1](#).

- b. If a problem has been created, list the incidents related to the specific problem using the `listIncidents` command, as described in [Section 12.4.2.2](#).
3. If an incident has not been created, go to Step 4. If an incident has been created, go to Step 5.
4. If you do not see any incidents listed that are related to your problem, you can create an incident manually using the `createIncident` command to capture diagnostics for the problem.

Consider creating an incident when you encounter an issue, such as software failure or performance problem, and you want to gather more diagnostic data. You can view the log files and the messages in the files. If there is a specific message that you believe is related to the issue you are seeing, you can use the message ID in the `createIncident` command.

See [Section 12.4.6.1](#) for more information about creating an incident.

5. View the details of the specific incident using the `showIncident` command, as described in [Section 12.4.2.2](#). This command lists information about the incident, including the related message ID, the time of the incident, the ECID, and the files generated by the incident.
6. Use the `getIncidentFile` command to view the contents of files for the incident, as described in [Section 12.4.2.2](#). The contents may provide information to guide you to the source of the problem and help in resolving it.
7. If the contents of the files for the incident do not help you to resolve the problem, you can execute additional dumps to view detailed diagnostics. For example, if you are experiencing performance problems, execute the `dms.metrics dump`. See [Section 12.4.4](#) for information about the dumps available and how to execute them.
8. If you still cannot resolve the problem, package the incident, along with the RDA report, and send them to Oracle Support. See [Section 12.4.6.3](#) and [Section 12.4.7](#) for information about packaging incidents and generating RDA reports.

12.4.2 Viewing Problems and Incidents

You can view the set of problems, the list of incidents, and the details of a particular incident using the WLST command-line utility, as described in the following topics:

- [Viewing Problems](#)
- [Viewing Incidents](#)
- [Querying Incidents](#)

12.4.2.1 Viewing Problems

You can view the set of problems by executing the WLST `listProblems` command, using the following format:

```
listProblems([adrHome]#[,server])
```

The `listProblems` command lists the problems in the ADR home. Each problem has a unique ID:

```
listProblems()
Problem#Id#####Problem#Key
#####1#####BEA-101020#[HTTP]
```


12.4.2.2 Viewing Incidents

You can list of all available incidents or the incidents related to a specific problem by executing the WLST `listIncidents` command, using the following format:

```
listIncidents([id],[ADHome])
```

For example, to see the list of all incidents, use the following command:

```
listIncidents()
Incident#Id#####Incident#Time#####Problem#Key
#####2#####Tue#May#28#11:05:59#PDT#2013#####MDS-50500#[MANUAL]
#####1#####Tue#May#28#11:02:22#PDT#2013#####MDS-50500#[MANUAL]
```

To view the incidents related to a specific problem, use the following command:

```
listIncidents(id='1')
Incident#Id#####Incident#Time#####Problem#Key
#####2#####Tue#May#28#11:05:59#PDT#2013#####MDS-50500#[MANUAL]
#####1#####Tue#May#28#11:02:22#PDT#2013#####MDS-50500#[MANUAL]
```

To view the details of a particular incident, use the WLST `showIncident` command, using the following format:

```
showIncident(id,[adrHome]#[,server])
```

For example, to see the details of incident 1, use the following command:

```
showIncident(id='1')
Incident#Id:#1
Problem#Id:#1
Problem#Key:#MDS-50500#[MANUAL]
Incident#Time:Tue#May#28#11:02:22#PDT#2013
Error#Message#Id:#MDS-50500
Execution#Context:
Flood#Controlled:#false
Dump#Files#:
###readme.txt
###jvm_threads10_i1.txt
###dms_metrics11_i1.txt
###dfw_samplingArchive13_i1.JVMThreadDump.txt
###dfw_samplingArchive13_i1.readme.txt
###odl_logs14_i1.txt
```

To view the contents of a file in the incident, use the WLST `getIncidentFile` command, using the following format:

```
getIncidentFile(id,#name#[,outputFile]#[,adrHome]#[,server])
```

For example, to view the contents for the file `odl_logs4_i1.dmp` use the following command:

```
getIncidentFile(id='1',#name='odl_logs14_i1.txt',outputFile='/tmp/odl_logs4_i1_
dmp.output')
```

The command writes the output to the file `odl_logs4_i1_dmp.output`.

12.4.2.3 Querying Incidents

While the `listIncidents` command shows you the incidents related to a particular problem ID, or for a particular server, it does not allow you to restrict the list further. The WLST `queryIncidents` command lets you query for the value of particular

attributes across one or more servers, or all servers in a domain. For example, you can query by the time of incident creation or the ECID.

An expression contains an incident attribute, an operator, and a string, in the following format:

```
attribute operator "string"
```

You can combine query expressions with the Boolean operators AND or OR, and group them by parentheses ().

The following incident attributes are supported:

- **TIMESTAMP:** Incident creation time. You can use the `from` and `to` operators to specify a time range. The date format is `YYYY-MM-DD#HH:MM`.
- **ECID:** Execution Context ID
- **PROBLEM_KEY:** Problem Key
- **MSG_FACILITY:** The error message facility, such as ORA or OHS.
- **MSG_NUMBER:** The error message ID, such as 600.

Custom incident attributes are also supported. For example, `TRACEID`, `APP`, `URI`, `AND` `DSID` are supported. In addition, the context values, as shown in the incident `readme.txt` file, are supported. For example, `DFW_APP_NAME` and `DFW_USER_NAME` are supported.

The following operators are supported:

- `equals`
- `notEqual`
- `startsWith`
- `endsWith`
- `contains`
- `isNull`
- `notNull`

For example, you can query all incidents in all servers in the domain for the ECID `f19wAgN000001`:

```
queryIncidents(query="ECID#equals#f19wAgN000001")
```

The following example queries all incidents that occurred between April 1, 2013 and April 15, 2013, for the server `wls_server_1`:

```
#queryIncidents(query="TIMESTAMP#from#'2013-04-01#00:00'AND#TIMESTAMP#to#  
'2013-04-15#00:00'",#  
servers=["wls_server_1"])
```

For more information about this command, see `queryIncidents` in the *WLST Command Reference for Infrastructure Components*.

12.4.3 Analyzing Specific Problem Keys

The Diagnostic Framework provides a set of well-defined problem keys for unhandled exceptions. These exceptions are either detected through the existing WLDF Watch "UncheckedException" or through the Diagnostic Framework `java.lang.Thread.UncaughtExceptionHandler` uncaught exception handler. Previously,

the Diagnostic Framework generated problem keys with different formats for the same type of issues. [Table 12-6](#) describes these problem keys and how to use them to investigate a problem.

Table 12-6 Uncaught Exception Problem Keys

Exception	Problem Key	Description
java.lang.OutOfMemoryError	DFW-99997 [java.lang.OutOfMemoryError]	Used by all java.lang.OutOfMemoryError incidents. With each incident of this type, a jvm.classhistogram dump is executed. The dump captures statistics about the instances of classes that have been loaded and the counts of associated Objects. Review the contents of this dump for a good starting point for understanding what has been loaded into the JVM's memory. In addition, the dms.metrics dump records statistics about the overall JVM memory.
java.sql.SQLException	DFW-99996 [ora-code java.sql.SQLException] [package.class.method][app-name]	Used for all exceptions of type java.sql.SQLException, including its subclasses. The Diagnostic Framework attempts to extract the Oracle error code from the exception error message, and if it is successful, uses that in the problem key. If not, it uses the exception name. Review the text associated with the exception to get more details, such as the operation that could not be performed on the database. In addition, you can review the SQL error code details for additional information.
All others	DFW-99998 [exception-name][package.class.name] [app-name]	Used by all other types of exceptions, such as java.lang.NullPointerException, java.io.IOException, java.lang.StringIndexOutOfBoundsException, that are not handled in a unique way. Review the text associated with the exception to get more details, such as the reason for the failure. The source line in the problem key is a best-attempt indicator of the location of the failure.

12.4.4 Working with Diagnostic Dumps

If you suspect a problem, you can make use of the built-in diagnostic dumps to report detailed diagnostics that can help diagnose the problem. Diagnostic dumps provide a means to output and record diagnostics data which serve as valuable information when diagnosing issues with Oracle Fusion Middleware components, applications, and infrastructure. The output from these dumps is intended to be used by customers and Oracle Support to diagnose issues with Oracle Fusion Middleware.

Diagnostic dumps are executed in the following ways:

- Manually, using WLST commands, as described in the following sections
For example, if your Java EE application is hanging and you suspect a deadlock, you could use the jvm.threads dump to obtain the set of threads.
- Automatically, when the Diagnostic Framework detects a critical error and creates an incident or when the administrator creates an incident

12.4.4.1 Listing Diagnostic Dumps

You can find a list of diagnostic dumps that are available for a Managed Server by executing the WLST `listDumps` command, using the following format:

```
listDumps([appName]#[,server])
```

For example, to list the available dumps for `wls_server1`:

```
listDumps(server='wls_server1')
Location#changed#to#domainRuntime#tree.#This#is#a#read-only#tree#with#DomainMBean#
as#the#root.#
For#more#help,#use#help(domainRuntime)
dfw.samplingArchive
dms.configuration
dms.ecidctx
dms.metrics
http.requests
jvm.classshistogram
jvm.threads
mds.MDSInstancesDump
odl.activeLogConfig
odl.logs
odl.quicktrace
opss.diagTest
opss.identityStoreUserRoleApiConfig
opss.securityContext
wls.image
#
Use#the#command#describeDump(name=<dumpName>)#for#help#on#a#specific#dump.
```

Table 12-7 lists the diagnostic dump actions that are defined by Oracle Fusion Middleware and their descriptions.

Table 12-7 Diagnostic Dump Actions

Dump Action	Description
dms.ecidctx	The data associated with a specific Execution Context ID (ECID), if specified. Otherwise, the data associated with all available ECIDs.
dms.metrics	Dynamic Monitoring Service (DMS) metrics. For information about these metrics, see "About Dynamic Monitoring Service (DMS)" in <i>Tuning Performance</i> .
http.requests	A summary of the currently active HTTP requests.
jvm.classshistogram	A JVM class histogram, the output of which varies depending on the JVM vendor.
jvm.flightRecording	The active JRockit Flight Recorder recording.
jvm.threads	Summary statistics about the threads running in a JVM as well as performing a full thread dump.
mds.MDSInstancesDump	Information about each MDS instance in the current JVM.
odl.activeLogConfig	The active Java logging configuration.
odl.logs	Contents of diagnostic logs, correlated by ECID or time range.
odl.quicktrace	Quick trace messages.
wls.image	The WLDF server image dump.

12.4.4.2 Viewing a Description of a Diagnostic Dump

You can view a description of a particular dump, including the syntax for executing the dump by using the WLST `describeDump` command. You specify the name of the dump in which you are interested. For example, to view a description of the `dms.metrics` dump, use the following command:

```
describeDump(name='dms.metrics')
Name:#dms.metrics
Description:#Dumps#DMS#(Dynamic#Monitoring#Service)#metrics.
Run#Mode:#asynchronous
Mandatory#Arguments:#
Optional#Arguments:
###Name#####Type#####Description
###dump#####STRING###How#much#to#dump
###servers#####STRING###Server#names
###names#####STRING###Name#of#DMS#noun#or#metric
###format#####STRING###Format#of#the#dump#output;#raw#or#xml
###nountypes###STRING###Type#of#DMS#noun
```

12.4.4.3 Executing Dumps

If you detect a problem and want to gather additional diagnostic data, you can invoke the `executeDump` command for a specified dump. Each dump may have mandatory or optional arguments, or both. To view the arguments for a particular dump and how to specify them, use the `describeDump` command, as described in [Section 12.4.4.2](#).

The following example executes the dump with the name `dms.metrics` and the incident ID 1 and writes it to the file `dumpout.txt`:

```
executeDump(name='dms.metrics',#outputFile='/tmp/dumpout.txt',#id='1')
Dump#file#dms_metrics1_11.dmp#added#to#incident#1
```

The command writes the dump output to the information about incident 1. If you execute the `showIncident` command for incident 1, the output includes `dms_metrics1_11.dmp`.

12.4.5 Configuring and Using Diagnostic Dump Sampling

Diagnostic dump sampling captures the output of diagnostic dumps at specified intervals. By sampling at regular intervals, diagnostic dump sampling can help to reveal issues such as slow running web requests, and where work is being performed in those requests.

This section contains the following topics:

- [Understanding Diagnostic Dump Sampling](#)
- [Configuring Dump Sampling](#)
- [Listing Dump Samplings](#)
- [Retrieving the Dump Sampling Output](#)

12.4.5.1 Understanding Diagnostic Dump Sampling

All diagnostic dump samplings are performed in the background, at specified intervals. By default, `jvm.threads` and `jvm.classhistogram` dumps are configured for sampling. However, they are not active until an incident is generated. Then, they remain active for 72 hours, by default.

You can modify the settings for the default dump samplings and you can create new sampling definitions for the dump actions listed in [Table 12-7](#) and for any

application-specific dumps. You can configure multiple sampling definitions for the same diagnostic dump, specifying different settings, such as sampling interval or server.

For each diagnostic dump sampling, the Diagnostic Framework stores the specified number of samples. When that limit is reached, the oldest sample is purged. All samples are purged when the server shuts down.

Table 12–8 shows the settings of the dump samplings that are configured by default.

Table 12–8 Default Diagnostic Dump Samplings Configuration

Dump Name	Sampling Interval	Maximum Samples Stored
jvm.threads	60 seconds	10
jvm.classhistogram	30 minutes	5

The Diagnostic Framework triggers the retrieval of the dump samples whenever an incident is created (through error detection or manual incident creation.) In addition, you can retrieve the contents of the dump samples, as described in Section 12.4.5.4.

You can retrieve the dump sample archives in either text or zip files:

- Text:** By default, the diagnostic dump samples are concatenated into a single archive file, in text format. An ASCII header and footer are wrapped around each sample in the archive file. The header contains a timestamp and the name of the diagnostic dump that produced the sample. Both the header and footer contain the number of the samples in the archive and the number of the particular sample. For example:

```

$$$===#BEGIN#OF#Diagnostic#Dump#-#jvm.classhistogram#(Archive##0#1_of_2)#===$$$
Fri#Sep#07#07:00:00#PDT#2013

<text#of#dump#sampling>#
#
$$$===#END#OF#Diagnostic#Dump#-#jvm.classhistogram#(Archive##0#1_of_2)#===$$$

```

- Zip:** You can configure diagnostic dump samplings to return a zip file instead of a concatenated file. The zip file contains all available dump sample files. This format supports any diagnostic dumps whose outputs are in binary format not suitable for concatenation, as well as for dumps that generate output in text format. This format also reduces the size of the archive containing the samples.

The following example shows the contents of a zip file:

```

unzip#-1#jvm_dump.zip
Archive:##jvm_dump.zip
##Length####Date###Time####Name
#-----#####-----#####-----
###508780##08-21-13#07:25###dfw_
samplingArchive1065570966467923683.JVMThreadDump.dmp
#####840##08-21-13#07:25###dfw_samplingsArchive7749640004639161119.readme.txt
#-----#####-----
###509620#####2#files

```

In addition to a text or zip file, when you retrieve a dump sample, the Diagnostic Framework generates a readme file. The readme file either lists the line numbers for each dump sample in the archive (for text format) or the individual sample file names (for zip format). It also lists the timestamp for each sample and the index for the archive.

The dump sample files are named using the following format:

```
dfw_dumpArchive $nnn$ .Sampling_Name.{txt|zip}
```

In the format nnn is a unique number assigned by the Diagnostic Framework.

For example, the following is an example of the name of a dump sample file for JVMThreadDump:

```
dfw_dumpArchive17394218037.JVMThreadDump.txt
```

The readme files are named using the following format:

```
dfw_dumpArchive $nnn$ .readme.txt
```

In the format nnn is a unique number assigned by the Diagnostic Framework.

All samplings are scheduled to begin at the next nearest interval, corresponding to the frequency. For example, if a sampling is configured at 12:05:13 PM and the frequency is 5 seconds, the sample will be collected at 12:05:15 PM. This ensures that the collection of a series of samplings with the same frequency will occur at the same time. It also aligns all samples across machines, assuming their system clocks are synchronized.

Note: You must be connected to the Administration Server to execute the WLST dump sampling commands.

12.4.5.2 Configuring Dump Sampling

You can create additional dump samplings, update existing dump samplings, remove dump samplings and enable or disable dump sampling, as described in the following topics:

- [Activating the Default Samples](#)
- [Creating Dump Samplings](#)
- [Modifying Dump Sampling Settings](#)
- [Removing Dump Samplings](#)
- [Enabling or Disabling All Dump Sampling](#)

12.4.5.2.1 Activating the Default Samples By default the `jvm.threads` and `jvm.classhistogram` dumps are not activated until an incident occurs. Then, they are active for 72 hours, by default.

You can change the behavior so that the dumps are active even if an incident has not occurred by setting the value of the MBean `DumpSamplingIdleWhenHealthy` to `false`.

To change the amount of time used for determining the system's health, change the value of the `DumpSamplingMinimumHealthyPeriod` MBean.

For information about changing the value of the Diagnostic Framework MBeans, see [Section 12.3.1](#)

12.4.5.2.2 Creating Dump Samplings You can create dump samplings for any dump listed in [Table 12-7](#) and for any application-specific dumps. To create dump samplings, use the WLST command `addDumpSample`. The `addDumpSample` command uses the following syntax:

```
addDumpSample(sampleName="sample_name", #diagnosticDumpName="dump_name",
#appName="application_name", ]#samplingInterval=num_seconds,
```

```
#rotationCount=num_samples,#[dumpedImplicitly={true|false},]
#[toAppend={true|false},]#[args={"arg_name":#"value"},]#
#[server="server_name"]
#
```

For example, to create a dump sampling for the `http.requests` dump, setting the sampling interval to 300 seconds and the rotation count to 10 samples, for the server `wls_server1`:

```
addDumpSample(sampleName="HTTPSampling",#diagnosticDumpName="http.requests",#
samplingInterval=300,#rotationCount=10,#server="wls_server1")
```

```
HTTPSampling#is#added
```

For complete syntax, see "addDumpSample" in the *WLST Command Reference for Infrastructure Components*.

12.4.5.2.3 Modifying Dump Sampling Settings You can change the settings of existing dump samplings by using the WLST command `updateDumpSample`. The `updateDumpSample` command uses the following syntax:

```
updateDumpSample(sampleName="sample_name",##
[#####appName="application_name",]#samplingInterval=num_seconds,#
#####rotationCount=num_samplings,#[dumpedImplicitly={true|false},]#
[#####toAppend={true|false},]#[args={"arg_name":#"value"},]#
[#####server="server_name"]
#
```

For example, to modify the dump sampling `HTTPSampling`, changing the sampling interval to 200 and the rotation count to 5:

```
updateDumpSample(sampleName="HTTPSampling",#samplingInterval=200,
#####rotationCount=5,#server="wls_server1")
```

```
HTTPSampling#is#updated
```

For complete syntax, see "updateDumpSample" in the *WLST Command Reference for Infrastructure Components*.

12.4.5.2.4 Removing Dump Samplings You can remove existing dump samplings using the WLST command `removeDumpSample`. The `removeDumpSample` command uses the following syntax:

```
removeDumpSample(sampleName="sample_name",#[server="server_name"]
#
```

For example, to remove the dump sampling `HTTPSampling`:

```
removeDumpSample(sampleName="HTTPSampling",#server="wls_server1")
```

```
Removed#HTTPSampling
```

For complete syntax, see "removeDumpSample" in the *WLST Command Reference for Infrastructure Components*.

12.4.5.2.5 Enabling or Disabling All Dump Sampling

You can enable or disable all dump sampling using the WLST command `enableDumpSampling`. This command affects all configured dump samplings. The `enableDumpSampling` command uses the following syntax:

```
enableDumpSampling(enable={true|false},#[server="server_name"])
```


Note that the server parameter is valid only if you are connected to the Administration Server. If you do not specify the server parameter, dump sampling is disabled for the Administration Server.

For example, to disable dump sampling for the Administration Server:

```
enableDumpSampling(enable=false)
```

```
Dump#sampling#disabled
```

To determine if dump sampling is enabled or disabled, use the WLST command `isDumpSamplingEnabled`. The `isDumpSamplingEnabled` command uses the following format:

```
isDumpSamplingEnabled([server="server_name"])
```

For complete syntax, see "enableDumpSampling" and "isDumpSamplingEnabled" in the *WLST Command Reference for Infrastructure Components*.

12.4.5.3 Listing Dump Samplings

You can list dump samplings using the WLST command `listDumpSamples`. You can list all dump samplings, a specified dump sampling, or all dump samplings associated with a specified server. The `listDumpSamples` command uses the following syntax:

```
listDumpSample([sampleName="sample_name"],#[server="server_name"])
#
```

For example, to list all dump samplings associated with the server `wls_server1`:

```
listDumpSamples(server="wls_server1")
Name#####:JVMThreadDump
Dump#Name#####:jvm.threads
Application#Name##:
Sampling#Interval#:#30
Rotation#Count####:#20
Dump#Implicitly###:#true
Append#Samples####:#true
Dump#Arguments####:#context=true,#timing=true,#progressive=true,#depth=20,#
threshold=30000
#
Name#####:JavaClassHistogram
Dump#Name#####:jvm.classshistogram
Application#Name##:
Sampling#Interval#:#1800
Rotation#Count####:#5
Dump#Implicitly###:#false
Append#Samples####:#true
Dump#Arguments####:#
```

For complete syntax, see "listDumpSample" in the *WLST Command Reference for Infrastructure Components*.

12.4.5.4 Retrieving the Dump Sampling Output

To retrieve the output of dump samples, you can use the WLST `executeDump` command or the WLST `getSamplingArchives` command, as described in the following topics:

- [Retrieving Dump Samples Using the executeDump Command](#)
- [Retrieving Dump Samples Using the getSamplingArchives Command](#)

12.4.5.4.1 Retrieving Dump Samples Using the executeDump Command You can retrieve dump samples using the WLST `executeDump` command, specifying the `dfw.samplingArchive` dump. This command collects all default sample archives and any dump samples that are specified with the parameter `dumpImplicitly=true` from a temporary location and concatenates them into a single file. The command also returns a readme file, with details of the dump samples.

When you use the `executeDump` command, you use the following syntax:

```
executeDump (name="dfw.samplingArchive", outputFile="filename" #
```

For the `outputFile` parameter, you can specify a text file or a zip file. If you specify a zip file, you must use the argument `zipOutput=true`.

For any dump sampling that is configured with the parameter `dumpImplicitly=false`, you must specify the optional `dfw.samplingArchive` argument `sampleName` to collect the contents of those dump samples. For example:

```
executeDump (name='dfw.samplingArchive', #args={'sampleName'#: #
'JavaClassHistogram'})
```

For more information, see "`executeDump`" in the *WLST Command Reference for Infrastructure Components*.

12.4.5.4.2 Retrieving Dump Samples Using the getSamplingArchives Command You can retrieve dump samples using the WLST `getSamplingArchives` command. This command collects all dump samples in a zip file containing the individual dump sample files and a readme file. This method is particularly useful in dealing with binary format dumps.

The `getSamplingArchives` command uses the following syntax:

```
getSamplingArchives ([sampleName="sample_name"] #[, outputFile="filename" #
[, server="server_name"])
```

For example to retrieve the dump samples for the sampling `JavaClassHistogram`, use the following command:

```
getSamplingArchives (sampleName="JavaClassHistogram", #
outputFile="/tmp/sampling.zip")
#
```

The following shows the contents of the zip file:

```
unzip#-l#/tmp/sampling.zip
Archive:##/tmp/sampling.zip
##Length####Date##Time####Name
#-----#####-----###-----
##6241768##04-27-13#11:19###dfw_
samplingArchive8680976839106379444.JavaClassHistogram.dmp
#####552##04-27-13#11:19###dfw_samplingArchive7861027727509995202.readme.txt
#-----#####-----
##6242320#####2#files
```

For complete syntax, see "`getSamplingArchives`" in the *WLST Command Reference for Infrastructure Components*.

12.4.6 Managing Incidents

The Diagnostic Framework stores incidents, whether they are created automatically or manually, and Oracle Fusion Middleware provides tools to help you process incident

reports and to package those incidents to send to Oracle Support. The following sections describe:

- [Creating an Incident Manually](#)
- [Creating an Aggregated Incident](#)
- [Packaging an Incident](#)
- [Generating an RDA Report](#)
- [Purging Incidents](#)

12.4.6.1 Creating an Incident Manually

System-generated problems—critical errors generated internally—are automatically added to the Automatic Diagnostic Repository (ADR). You can gather additional diagnostic data on these problems, upload diagnostic data to Oracle Support, and in some cases, resolve the problems, all with the workflow that is explained in [Section 12.4](#).

Consider creating an incident manually when you encounter an issue, such as software failure or performance problem and you want to gather more diagnostic data, but the Diagnostic Framework has not automatically created an incident.

You use the WLST command `createIncident` to create an incident manually. You can specify an incident based on time, a message ID, an impact area, or an ECID. Then, you can inspect the content of the incident or send it to Oracle Support for further analysis.

For example, to manually create an incident based on a message ID:

1. Search the log files, as described in [Section 11.3.2](#). If you find a message that you suspect is related to the issue you are seeing, you can use the message ID when you create the incident.
2. Use the following commands to invoke WLST, connect to the Managed Server and navigate to the Managed Server instance:

```
java#weblogic.WLST
connect('username',#'password',#'localhost:7001')
cd('servers/server_name')
```

3. Create the incident, using the `createIncident` command, with the following format:

```
createIncident([adrHome]#[,incidentTime]#[,messageId]#[,ecid]#[,appName]
##[,description]#[,server])
```

For example, to create an incident based on the error with the message ID MDS-50500, use the following command, specifying the message ID, and provide a description of the incident to help you and Oracle support track the incident:

```
createIncident(messageId='MDS-50500',#description='sample#incident')
Incident#Id:#1
Problem#Id:#1
Problem#Key:#MDS-50500#[MANUAL]
Incident#Time:Tue#May#28#11:02:22#PDT#2013
Error#Message#Id:#MDS-50500
Execution#Context:null
Flood#Controlled:#false
Dump#Files#:
###jvm_threads10_i1.txt
###dms_metrics11_i1.txt
```

```
###dfw_samplingArchive13_i1.JVMThreadDump.txt
###dfw_samplingArchive13_i1.readme.txt
###odl_logs14_i1.txt
```

If you do not specify a server, the incident collects information from the server to which you are connected. To specify a server, use the `server` option, as shown in the following example:

```
createIncident (messageId='MDS-50500', #description='sample#incident', #
server='wls_server1'
)
```

If you do not specify the `adrHome` option, the incident is created in the server to which you are connected. For example, if you are connected to the Administration Server, the incident is created in the `adrHome` for the Administration Server.

The Diagnostic Framework evaluates the command and invokes the appropriate diagnostic dumps. The incident and the diagnostic dumps are written to the ADR. Each diagnostic dump writes its output to the incident.

You can view the information about the incident, as described in [Section 12.4.2.2](#).

You can view the information in the dumps, as described in [Section 12.4.4](#).

12.4.6.2 Creating an Aggregated Incident

If you have several incidents and want to combine them into a single incident, you can use the WLST `createAggregatedIncident` command. For example, if you used selective tracing, the resulting incidents containing the trace data may be generated on multiple servers. With the `createAggregatedIncident` command, you can generate an aggregated incident that meets criteria you specify. The original incidents are untouched. That is, the aggregated incident contains a copy of the incident files from the queried incidents.

The aggregated incidents are created on the Administration Server host, but they can contain incidents from one or more servers or all servers in the domain.

You construct a query using an expression that contains an incident attribute, an operator, and a string, in the following format:

```
attribute operator "string"
```

You can combine query expressions with the Boolean operators AND or OR, and group them by parentheses ().

For information about the supported attributes and operators, see "createAggregatedIncident" in the *WLST Command Reference for Infrastructure Components*.

Each aggregated incident will contain a zip file for each incident returned from the query, as well as descriptive text detailing the query used and the details of each incident.

For example, to create an aggregated incident for all incidents that contain the ODL_TRACE_ID of 123456 on the server `wls_server1`:

```
createAggregatedIncident (query="ORDL_TRACE_ID#equals#123456", #servers="wls_
server1")
Incident#55#created, #containing#the#following#incidents:
Server#wls_server1
Incident#Id####Problem#Key#####Incident#Time
15#####TRACE#[123456]#[MANUAL]#####Mon#Apr#15#11:22:12#EDT#2013
```

To create an aggregated incident for all incidents that contain the ODL_TRACE_ID of 123456 on all servers in the domain:

```
createAggregatedIncident(query="ORDL_TRACE_ID#equals#123456")
Incident#55#created,#containing#the#following#incidents:
Server#wls_server1,#wls_server2
Incident#Id###Problem#Key#####Incident#Time
15#####TRACE#[123456]#[MANUAL]#####Mon#Apr#15#11:22:12#EDT#2013
```

12.4.6.3 Packaging an Incident

You can package the incident to facilitate sending the information to Oracle Support by using the ADR Command Interpreter (ADRCI). The ADRCI utility enables you to investigate and report problems in a command-line environment. With ADRCI, you can package incident and problem information into a zip file for transmission to Oracle Support.

The ADRCI command-line utility is located in the following directory:

```
(UNIX)#ORACLE_HOME/wlserver/server/adr
(Windows)#ORACLE_HOME\wlserver\server\adr
```

Packaging an incident involves a three-step process:

1. Create a logical package.

The package is denoted as logical because it exists only as metadata in the ADR. It has no content until you generate a physical package from the logical package. The logical package is assigned a package number, and you refer to it by that number in subsequent commands.

You can create the logical package as an empty package, or as a package based on an incident number, a problem number, a problem key, or a time interval. If you create the package as an empty package, you can add diagnostic information to it in step 2.

Creating a package based on an incident means including diagnostic data, such as dumps, for that incident. Creating a package based on a problem number or problem key means including in the package diagnostic data for incidents that reference that problem number or problem key. Creating a package based on a time interval means including diagnostic data on incidents that occurred in the time interval.

2. Add diagnostic information to the package.

If you created a logical package based on an incident number, a problem number, a problem key, or a time interval, this step is optional. You can add additional incidents to the package or you can add any file within the ADR to the package. If you created an empty package, you must use ADRCI commands to add incidents or files to the package.

3. Generate the physical package.

When you submit the command to generate the physical package, ADRCI gathers all required diagnostic files and adds them to a zip file in a designated directory. You can generate a complete zip file or an incremental zip file. An incremental file contains all the diagnostic files that were added or changed since the last zip file was created for the same logical package. You can create incremental files only after you create a complete file, and you can create as many incremental files as you want. Each zip file is assigned a sequence number so that the files can be analyzed in the correct order.

Zip files are named according to the following format:

```
packageName_mode_sequence.zip
```

In the format:

- `packageName` consists of a portion of the problem key followed by a timestamp.
- `mode` is either `COM` or `INC`, for complete or incremental.
- `sequence` is an integer.

For example, to package an incident, take the following steps:

1. Set the `ORACLE_HOME` and `LD_LIBRARY_PATH` environment variables as shown in the following example:

```
ORACLE_HOME=ORACLE_HOME/oracle_common  
LD_LIBRARY_PATH=ORACLE_HOME/wlserver/server/adr#
```

2. Invoke `ADRCI`. For example:

```
ORACLE_HOME/wlserver/server/adr/adrci
```

3. Use the `SET BASE` command to specify the ADR Base and the `SET HOMEPATH` command to specify the ADR home that contains the incident. The path for the `HOMEPATH` is relative to the ADR Base.

```
SET#BASE#/scratch/oracle/config/domains/wls_domain/servers/wls_server1/adr  
SET#HOMEPATH#diag/ofm/wls_domain/wls_server1
```

4. Generate the logical package:

```
IPS#CREATE#PACKAGE#INCIDENT#incident_number
```

For example, the following command creates a package based on incident 1:

```
IPS#CREATE#PACKAGE#INCIDENT#1  
Created#package#1#based#on#incident#id#1,#correlation#level#typical
```

`ADRCI` assigns the logical package a number.

5. Optionally, you can add diagnostic information to the logical package. You can add the following types of information:
 - All diagnostic information for a particular incident. For example, you can add another incident that you think might be related to the incident you are packaging, using the following command:

```
IPS#ADD#INCIDENT#incident_number#PACKAGE#package_number
```

- A named file within the ADR. For example, if an incident is related to an application, you can add the `.ear` file for the application. You can also add a `readme` file with notes you provide to Oracle Support. For example, to add a file to the package, use the following command:

```
IPS#ADD#FILE#filespec#PACKAGE#package_number
```

6. Generate the physical package using the following command:

```
IPS#GENERATE#PACKAGE#package_number#IN#path
```

For example, to generate a package with the number 1, use the following command:

```
IPS#GENERATE#PACKAGE#1#in#/tmp#
Generated#package#1#in#file#/tmp/BEA337Web_20100223132315_COM_1.zip,#mode#
complete
```

This generates a complete physical package (zip file) in the designated path.

For more information, see the "ADRCI: ADR Command Interpreter" chapter of the *Oracle Database Utilities*.

12.4.6.4 Purging Incidents

By default, incidents are purged when the total size of all incidents exceed 500 MB. You can use the `maxTotalIncidentSize` MBean parameter to change this value, as described in [Section 12.3.1](#).

You can manually purge incidents using the ADRCI command. You can purge based on an ID or range of IDs, the age of the incident, or the type of incident. For example, to purge incidents that are older than 60 minutes, use the following command:

```
purge#-age#60#
```

See the "ADRCI: ADR Command Interpreter" chapter of the *Oracle Database Utilities*.

12.4.7 Generating an RDA Report

You can use the Remote Diagnostic Agent (RDA), a command-line diagnostic tool, to provide a comprehensive picture of your environment. Additionally, RDA can provide recommendations on various topics, for example configuration and security. This aids you and Oracle Support in resolving issues.

RDA is a set of command line diagnostic scripts that are executed by an engine written in the Perl programming language. RDA is used to gather detailed information about an Oracle environment; the data gathered is in turn used to aid in problem diagnosis. The output is also useful for seeing the overall system configuration.

RDA is designed to be as unobtrusive as possible; it does not modify systems in any way. A security filter is provided if required.

RDA collects information that is useful for troubleshooting issues in the following areas:

- Installation and configuration
- Performance
- ORA-600, ORA-7445, ORA-3113, and ORA-4031 errors
- Upgrade, migration, and linking
- Oracle Database
- Oracle Fusion Middleware

To run RDA, execute the following:

```
(UNIX)#ORACLE_HOME/oracle_common/rda.sh
(Windows)#ORACLE_HOME\oracle_common\rda.cmd
```

The following shows a part of the output:

```
#./rda.sh
-----
S000INI:#Initializes#the#Data#Collection
-----
```

```

RDA#uses#the#output#file#prefix#to#identify#all#files#belonging#to#the#same
data#collection.#The#prefix#must#start#with#a#letter#and#must#contain#only
alphanumeric#characters.
#
Enter#the#prefix#to#be#used#for#all#the#generated#files
Hit#'Return'#to#accept#the#default#(RDA)
>#
#
Enter#the#directory#used#for#all#the#files#to#be#generated
Hit#'Return'#to#accept#the#default
(/scratch/oracle1/Oracle/Middleware/Oracle_Home/oracle_common/rda/output)
>#
#
Do#you#want#to#keep#report#packages#from#previous#runs#(Y/N)?
Hit#'Return'#to#accept#the#default#(N)
>#
#
Enter#the#Oracle#home#to#be#used#for#data#analysis
Hit#'Return'#to#accept#the#default
(/scratch/oracle1/Oracle/Middleware/Oracle_Home)
For more information about RDA, see the readme file, which is located at:

(UNIX) #ORACLE_HOME/oracle_common/rda/README_Unix.txt
(Windows) #ORACLE_HOME\oracle_common\rda\README_Windows.txt

```

12.5 Managing and Running the Health Test Framework

Oracle Fusion Middleware provides a Health Test Framework that includes diagnostic tests that are designed to exercise particular aspects of Oracle Fusion Middleware and its applications, to determine whether they are operating correctly and to help identify and resolve any problems.

This section contains the following topics:

- [Understanding the Health Test Framework](#)
- [Understanding the Health Test Framework File Repository](#)
- [Using the Health Test Framework Command Line](#)
- [Managing the Health Test Framework](#)
- [Running Health Test Framework Diagnostic Tests](#)
- [Searching for Health Test Framework Diagnostic Tests](#)
- [Retrieving a Description of a Health Test Framework Test](#)
- [Listing Health Test Framework Test Runs](#)
- [Generating Health Test Framework Reports](#)

12.5.1 Understanding the Health Test Framework

The Health Test Framework lets you execute diagnostic tests and collects the results into detailed diagnostic reports. The diagnostics tests are installed when you install or patch Oracle Fusion Middleware.

You can use the Health Test Framework to check normal system health and to troubleshoot system problems. You can configure your Oracle Fusion Middleware environment to run the diagnostic tests using the Health Test Framework command line interface.

A diagnostic test may or may not be associated with a particular error message. If an Oracle Fusion Middleware application handles a particular error in a way that triggers the creation of an incident, then any diagnostic tests that are associated with the error message for the incident run automatically. The test results are associated with the incident and the identity of the user who received the error message is recorded. The diagnostic tests and results are stored in file-based repository, as described in [Section 12.5.2](#).

12.5.2 Understanding the Health Test Framework File Repository

The Health Test Framework stores tests and test results in a file-based repository. The repository has two primary stores:

- TestDef, which contains test definitions, test metadata, data related to test parameters, language data files, and index files.
- Test Run, which contains data related to the test run and execution, report files, and index files.

Index files are used for improving the speed of searching and querying.

The repositories are created when you register tests or run tests if the repository does not exist at the specified location.

You can specify the location of the repository in the configuration properties file, `diagconfig.properties`. This file is located in:

```
MW_HOME/diagbase/dtf_filestore
```

By default, the file stores are located in:

```
MW_HOME/diagbase/dtf_filestore/testdef
MW_HOME/diagbase/dtf_filestore/testrun
```

12.5.3 Using the Health Test Framework Command Line

The Health Test Framework provides two command line interfaces, `dfwhealthtestadminctl.sh` for administration commands and `dfwhealthtestctl.sh` for execution commands, as described in the following sections:

- [dfwhealthtestadminctl.sh Command Line](#)
- [dfwhealthtestctl.sh Command Line](#)

12.5.3.1 dfwhealthtestadminctl.sh Command Line

The Health Test Framework provides the `dfwhealthtestadminctl.sh` command line for administering the health tests. With it, you can register tests and rebuild the index.

Before you can run the `dfwhealthtestadmin.ctl` command, you must set the following environment variables:

- `ORACLE_HOME`: The Oracle home
- `JAVA_HOME`: The Java home
- `dtf_fs_diagbase`: The location of the repository

The `dfwhealthtestadminctl.sh` command-line interface is located in:

```
ORACLE_COMMON_HOME/common/bin
```

The syntax of the command-line interface is:

`/dfwhealthtestadminctl.sh#command#[options]`

Table 12–9 lists the Health Test Framework administration commands:

Table 12–9 Health Test Framework `dfwhealthtestadminctl.sh` Commands

Command	Description
help	Provides command line help.
register	Registers one or more Health Test Framework test with the repository.
index	Rebuilds the index with the existing data files.

12.5.3.1.1 help Provides help for the commands.

The syntax is:

`help#command`

The following table describes the parameters for the help command:

Parameter	Description
command	The name of the command.

12.5.3.1.2 register Registers one or more tests with the repository.

The syntax is:

```
register#testfile=test_xml_files#|#dir=test_dirs
##### [validateonly={Y|N}]
#####testjar=jar_file_location
```

The following table describes the parameters for the register command:

Parameter	Description
testfile	One or more XML test files to register. Provide the path to the files, using a comma-separated list for more than one file.
dir	One or more directories to register. Provide the path to the directories, using a comma-separated list for more than one directory. Use this option to register multiple tests from the same directory.
validateonly	Optional. A Boolean value to specify whether to only validate the test or to upload the test or tests to the repository. Value values are Y (to only validate the test) or N (to upload the tests).
testjar	The location of the test jar files.

12.5.3.1.3 index Rebuilds the index for the testdef or testrun repository, using the data files in the repository.

The syntax is:

`index#[refresh={testdef|testrun}]`

The following table describes the parameters for the index command:

Parameter	Description
refresh=testdef	Refreshes the index for the testdef repository.
refresh=testrun	Refreshes the index for the testrun repository.

12.5.3.2 dfwhealthtestctl.sh Command Line

The Health Test Framework provides the `dfwhealthtestctl.sh` command line for running the health tests. With it, you can run tests, list runs and tests, get a description of a test, get the status of tests, and retrieve a report of a test or test run.

The `dfwhealthtestctl.sh` command-line interface is located in:

```
ORACLE_COMMON_HOME/common/bin
```

The syntax of the command-line interface is:

```
/dfwhealthtestctl.sh#command#[options]
```

Table 12–10 lists the Health Test Framework `dfwhealthtestctl.sh` commands:

Table 12–10 Health Test Framework `dfwhealthtestctl.sh` Commands

Command	Description
<code>desctest</code>	Provides a detailed description of a specified test.
<code>listrun</code>	List the test runs.
<code>listtest</code>	Lists the description of one or more tests.
<code>report</code>	Extract a report of a test run.
<code>run</code>	Run one or more tests.
<code>status</code>	Get the status of a test run or execution.

12.5.3.2.1 `desctest` Provides a detailed description of a specified test.

The syntax is:

```
desctest#{testid=testid#|#testname=testname}]
#####[showparam={Y|N}]
```

The following table describes the parameters for the `desctest` command:

Parameter	Description
<code>testid</code>	The ID of the test. To obtain the ID, use the <code>listtest</code> command. You must specify either this parameter or the <code>testname</code> parameter.
<code>testname</code>	The name of the test. You must specify either this parameter or the <code>testid</code> parameter.
<code>showparam</code>	Optional. A Boolean value to specify whether to retrieve the parameters for the test. Value values are Y or N.

12.5.3.2.2 `help` Provides help for the commands.

The syntax is:

```
help#command
```

The following table describes the parameters for the `help` command:

Parameter	Description
command	The name of the command.

12.5.3.2.3 listrun Retrieves summary information for the test runs. You can retrieve the information by run name, run status, start time, and test name.

The syntax is:

```
listrun#{#runname=runname#|#testname=testname#|#status=status#|#lasthours=hours#}
#####[showexec={Y|N}]
```

The following table describes the parameters for the listrun command:

Parameter	Description
runname	The name of the run for which you want to return results. To get the runname, use the listrun command without any parameters.
testname	The name of the test for which you want to return results.
status	The status of the run for which you want to return results. Valid values are running, warning and success.
lasthours	The number of hours for which you want to return results.
showexec	Optional. A Boolean value to specify whether to retrieve the execution of the tests. Value values are Y or N.

12.5.3.2.4 listtest Lists one or more tests. You can use wildcards to specify the names of the tests

The syntax is:

```
listtest#[testname=testname]
#####[productcode=productcode]
```

The following table describes the parameters for the help command:

Parameter	Description
testname	The names of one or more tests, in a comma-separated list. You can use wildcards to specify a pattern to match the names of tests.
productcode	The name of the product. You can use wildcards to specify a pattern to match the name of the product.

12.5.3.2.5 report Extract a test report for a particular test run or execution.

The syntax is:

```
report#{#runname=runname#|#runid=run_id#|#execid=execution_id#}
#####[destdir=destination_directory]
#####[format=HTML#|#XML]
#####[translate={Y#|#N}]
```

The following table describes the parameters for the report command:

Parameter	Description
runname	The name of the run. To get the name, use the listrun command. You must specify this parameter, runid, or execid.

Parameter	Description
runid	The ID of the run. To get the ID, use the <code>listrun</code> command. You must specify this parameter. <code>runname</code> , or <code>execid</code> .
execid	The execution ID of the run. To get the ID, use the <code>listrun</code> command. You must specify this parameter, <code>runname</code> , or <code>runid</code> .
destdir	Optional. The destination directory to which the report files are written. If not specified, the report files are extracted under the <code>java.io.tmpdir/user.name/diagfwk</code> directory, where <code>java.io.tmpdir</code> and <code>user.name</code> are Java system properties.
format	The format of the generated report. Valid values are <code>HTML</code> and <code>XML</code> .
translate	A Boolean value specifying whether to translate the report. Value values are <code>Y</code> or <code>N</code> .

12.5.3.2.6 run Run the specified tests.

```
run#{test=testnames#|#productcode=codes#}
#####[runname=name
#####[runoption=asynch]
#####[input:param1=value1#param2=value2 ...
#####[inputfile=filename]#
#####[contextfile=context_file]#
#####[moninterval=monitoring_interval]#
#####[nthreads=number_of_threads]#
#####[reportshowparam=show_param]#
```

The following table describes the parameters for the run command:

Parameter	Description
test	The names of one or more tests, in a comma-separated list. If you specify a single test name, you can also specify one or more input parameter names and values using the input parameter. You must specify this parameter or the productcode parameter.
productcode	The code for one or more products that include the tests you want to run. Specify more than one product code in a comma-separated list.
runname	Optional. A name for the current run. If not specified, the command generates a default run name.
moninterval	The interval, in seconds, in which the status of the run is uploaded to the test repository. The default value is 30 seconds.
nthreads	The number of parallel threads that should be spawned to execute tests in this run. The default value is 5. Specifying a value of 1 for this parameter will execute the tests serially.
runoption	Optional. Options for running the tests. The valid value is <code>asynch</code> , which runs the test asynchronously in the background.
input	Optional. If you use this parameter, use one or more parameter name/value pairs. For example: <code>domainhome=/scratch/oracle/domains/basedomain</code> You can only use this parameter when you run only one test.
inputfile	The path to a file containing input parameters. The file should have each input name/value pair on a separate line. The format of each line is: <code>input_param_name:input_param_value</code>

Parameter	Description
contextfile	The path to a file that contains user system properties to be used at run time.
reportshowparam	Sets the input and output parameter display. By default all parameters are hidden during execution. To specify that the parameters are shown, use input or output, or a comma-separated string with both. For example: reportshowparam=input,output

12.5.3.2.7 status Get the status of a test run by specifying the run name, run ID, or execution ID.

The syntax is:

```
status# {#runname=runname#|#runid=run_id#|#execid=execution_id#}
##### [printtree={Y|N}]
```

The following table describes the parameters for the status command:

Parameter	Description
runname	The name of the run. To get the run name, use the listrun command.
runid	The ID of the run. To get the ID, use the listrun command.
execid	The execution ID of the run. To get the ID, use the listrun command.
printtree	Specifies whether to print the status of all nested conditions. Valid values are Y and N.

12.5.4 Managing the Health Test Framework

You can register tests and rebuild the index, as described in the following topics:

- [Creating a Repository and Registering Health Test Framework Tests](#)
- [Rebuilding the Health Test Framework Indexes](#)

12.5.4.1 Creating a Repository and Registering Health Test Framework Tests

To create a repository, you register a test using the `dfwhealthtestadminctl.sh register` command. The command creates a repository and registers one or more tests with the repository.

To create a repository and register the tests, take the following steps:

1. Set the `dtf_fs_diagbase` environment variable to specify the location of the repository. For example:

`setenv##dtf_fs_diagbase#/scratch/Oracle/Middleware/diag`
2. Execute the following command, which creates the repository and registers one test in the repository:

```
dfwhealthtestadminctl.sh#register#testfile=/scratch/tests/sampleTest.xml#
#####testjar=/scratch/tests/testjar
```

You can register more than one test at a time using the `testfile` or `dir` parameters:

- To register more than one test using the `testfile` parameter, you can use a comma-separated list or wildcards.

The following example registers two tests, using a comma-separated list:

```
dfwhealthtestadminctl.sh#register#testfile=/scratch/tests/sampleTest.xml,##
#####/scratch/moretests/sampleTest.xml
#####testjar=/scratch/tests/testjar
```

The following example registers more than one test, using wildcards:

```
dfwhealthtestadminctl.sh#register#testfile=/scratch/tests/%Test.xml,##
#####testjar=/scratch/tests/testjar
```

- To register more than one test using the `dir` parameter, you can specify one or more directories in a comma-separated list.

The following example registers all tests in the specified directory:

```
dfwhealthtestadminctl.sh#register#dir=/scratch/dfw_tests##
```

12.5.4.2 Rebuilding the Health Test Framework Indexes

Indexes improve the speed of searching and querying the Health Test Framework repository. Indexes related to test definition are stored in the `testdef` file store. Indexes related to test runs and execution are stored in the `testrun` file store.

In some circumstances, you may want to refresh the indexes. You use the `dfwhealthtestadminctl.sh index` command, specifying the `testdef` or `testrun` file store. The following example refreshes the `testdef` file store:

```
dfwhealthtestadminctl.sh#index#refresh=testdef#
```

12.5.5 Running Health Test Framework Diagnostic Tests

You run the Health Test Framework diagnostic tests using the `dfwhealthtestctl.sh` command line interface, which is located in:

```
ORACLE_COMMON_HOME/common/bin
```

To run a test, use the `run` command, using the format described in [Section 12.5.3.2.6](#). For example, to run the test `SampleTest`, use the following command:

```
./dfwhealthtestctl.sh#command
#####[testfile=SampleTest.xml]#
```

You can run one or more tests at the same time:

- To run one or more tests by specifying the test names in a comma-separated list:

```
./dfwhealthtestctl.sh#run#test=oracle.dfw.healthtest.test.sample.SampleTest2
Processing#"run"#command#...
#
Executing#run#ID#"1340125981427"#with#name#"TestRun_1340125981427"#...
sampleTest2#sleep#for#a#while...
sampleTest2#done
#
Run#ID#"1340125981427"#with#name#"TestRun_1340125981427"#started#at#06/21/2013
#10:13:01#AM#and#completed#at#06/21/2013#10:13:04#AM.#No#diagnostic#issues#were#
detected.
```

- To run one or more tests by using wildcards for the test name:

```
./dfwhealthtestctl.sh#run#test=oracle.dfw.healthtest.test.sample.SampleTest*
```

- To run all tests associated with a particular product:

```
./dfwhealthtestctl.sh#run#productcode=idm
```

If you are running only one test, you can specify input options either on the command line or in an input file:

- To specify the input options on the command line, use the `input:` parameter:

```
./dfwhealthtestctl.sh#run#
test=oracle.dfw.healthtest.test.sample.SampleTest,oracle.dfw.healthtest.test.sa
mple.SampleTest2
#####input:userid=11#input:roleid=22
```

- To specify the input options in an input file, create the file and specify it on the command line using the `inputfile` parameter. The following shows the format of the input file:

```
input_parameter1:parameter_value1
input_parameter2:parameter_value2
```

For example:

```
userid:11
roleid:22
```

Then, specify the file on the command line:

```
./dfwhealthtestctl.sh#run#test=oracle.dfw.healthtest.test.sample.SampleTest
#####inputfile=/tmp/inputfile
```

12.5.6 Searching for Health Test Framework Diagnostic Tests

You can search for tests or runs using the `dfwhealthtestctl.sh listtest` command. Optionally, you can specify the name of the test or product code:

- To search for all tests:


```
./dfwhealthtestctl.sh#listtest
```
- To search for one test named `SampleTest`:


```
./dfwhealthtestctl.sh#listtest#testname=SampleTest
```
- To search for more than one test using wildcards:


```
./dfwhealthtestctl.sh#listtest#testname="S*"
```
- To search for all tests related to a particular product:


```
dfwhealthtestctl.sh#listtest#productcode=idm
```

12.5.7 Retrieving a Description of a Health Test Framework Test

You can retrieve the description of a specified test using the `dfwhealthtestctl.sh listtest` command. For example, to get the description of the test `SampleTest`, along with its parameters, use the following command:

```
./dfwhealthtestctl.sh#desctest#testname="SampleTest"#showparam=Y
```

12.5.8 Listing Health Test Framework Test Runs

You can query the results of test runs, by test name, run name, status, or time, using the `dfwhealthtestctl.sh listrun` command:

- To query the results for a test run by specifying the test name:
`./dfwhealthtestctl.sh#listrun#testname="SampleTest"`
- To query the results for a test run by specifying the run name:
`./dfwhealthtestctl.sh#listrun#runname="run_1" #`
- To query the results for a test run by specifying the status. For example, to get the results of all tests with a status of running:
`./dfwhealthtestctl.sh#listrun#status=r#`
- To query the results for test runs that were started in the last 2 hours:
`./dfwhealthtestctl.sh#listrun#lasthours=2#`

12.5.9 Generating Health Test Framework Reports

You can generate reports of the test runs for the Health Test Framework using the `dfwhealthtestctl.sh` report command. You can generate HTML or XML reports.

You can specify the run name, run ID or execution ID>

For example, to generate a report in HTML format for the run with the run ID of 1330128064268:

```
dfwhealthtestctl.sh#report#runid=1330128064268#format=HTML
```


Part VI

Advanced Administration

This part describes advanced administration tasks, such as managing the metadata repository and changing the network configuration, that involve reconfiguring Oracle Fusion Middleware.

Part VI contains the following chapters:

- [Chapter 13, "Managing the Metadata Repository"](#)
- [Chapter 14, "Changing Network Configurations"](#)

Managing the Metadata Repository

Many Oracle Fusion Middleware components use metadata repositories to hold configuration information about the component and metadata for applications. This chapter provides information on managing the metadata repositories used by Oracle Fusion Middleware.

It contains the following sections:

- [Understanding a Metadata Repository](#)
- [Creating a Database-Based Metadata Repository](#)
- [Managing the MDS Repository](#)
- [Managing Metadata Repository Schemas](#)
- [Purging Data](#)

13.1 Understanding a Metadata Repository

A metadata repository contains metadata for Oracle Fusion Middleware components, such as Oracle Application Development Framework. It can also contain metadata about the configuration of Oracle Fusion Middleware and metadata for your applications.

Oracle Fusion Middleware supports multiple repository types. A repository type represents a specific schema or set of schemas that belong to a specific Oracle Fusion Middleware component (for example, Oracle Application Development Framework.) Oracle Fusion Middleware supports Edition-Based Redefinition (EBR), which enables you to upgrade the database component of an application while it is in use, thereby minimizing or eliminating down time. The schemas in a repository can be EBR-enabled schemas.

A particular type of repository, the Oracle Metadata Services (MDS) Repository, contains metadata for certain types of deployed applications. This includes custom Java EE applications developed by your organization and some Oracle Fusion Middleware component applications. For information related specifically to the MDS Repository type, see [Section 13.3](#).

You can create a database-based repository or, for MDS, a database-based repository or a file-based repository. For production environments, you use a database-based repository. Some components require that a schema be installed in a database, necessitating the use of a database-based repository. MDS supports Edition-Based Redefinition (EBR) enabled schemas.

Note: After the database for the metadata repository has been used for the Oracle Fusion Middleware installation, the database, service name, or SID cannot be changed.

13.2 Creating a Database-Based Metadata Repository

You use the Oracle Fusion Middleware Metadata Repository Creation Utility (RCU) to create the metadata repository in an existing database. You can use RCU to create the MDS Repository or a repository for metadata for particular components. RCU creates the necessary schemas for the components. See *Creating Schemas with the Repository Creation Utility* for a list of the schemas and their tablespaces and datafiles.

With RCU, you can also drop component schemas.

For information about the supported databases and the supported versions, as well as using these databases with the MDS Repository, see "Supported Databases for the MDS Schema" in the *Oracle Fusion Middleware System Requirements and Specifications*.

Note: Oracle recommends that all metadata repositories reside on a database at the same site as the components to minimize network latency issues.

For information about managing an MDS Repository, see [Section 13.3](#).

See Also: *Creating Schemas with the Repository Creation Utility* for information about how to use RCU to create a database-based metadata repository

13.3 Managing the MDS Repository

Oracle Metadata Services (MDS) Repository contains metadata for certain types of deployed applications. Those deployed applications can be custom Java EE applications developed by your organization and some Oracle Fusion Middleware component applications, such as Oracle Web Services Manager. A Metadata Archive (MAR), a compressed archive of selected metadata, is used to deploy metadata content to the MDS Repository, which contains the metadata for the application.

You should deploy your applications to MDS in the following situations, so that the metadata can be managed after deployment:

- The application contains seeded metadata packaged in a MAR.
- You want to enable user personalizations at run time.

The following topics provide information about the MDS Repository:

- [Understanding the MDS Repository](#)
- [Registering and Deregistering a Database-Based MDS Repository](#)
- [Registering and Deregistering a File-Based MDS Repository](#)
- [Changing the System Data Source](#)
- [Using System MBeans to Manage an MDS Repository](#)
- [Viewing Information About an MDS Repository](#)
- [Configuring an Application to Use a Different MDS Repository or Partition](#)

- [Moving Metadata from a Source System to a Target System](#)
- [Moving from a File-Based Repository to a Database-Based Repository](#)
- [Deleting a Metadata Partition from a Repository](#)
- [Purging Metadata Version History](#)
- [Managing Metadata Labels in the MDS Repository](#)

See Also: *High Availability Guide* for information about using an MDS Repository with Oracle Real Application Clusters (Oracle RAC)

13.3.1 Understanding the MDS Repository

The MDS framework allows you to create customizable applications. A customized application contains a base application (the base documents) and one or more layers containing customizations. MDS stores the customizations in a metadata repository and retrieves them at run time to merge the customizations with the base metadata to reveal the customized application. Since the customizations are saved separately from the base, the customizations are upgrade safe; a new patch to base can be applied without breaking customizations. When a customized application is launched, the customization content is applied over the base application.

A customizable application can have multiple customization layers. Examples of customization layers are *industry* and *site*. Each layer can have multiple customization layer values, but typically only one such layer value from each layer is applied at run time. For example, the industry layer for a customizable application can contain values for health care and financial industries; but in the deployed customized application, only one of the values from this layer is used at a time. For more information about base documents and customization layers, see "Customizing Applications with MDS" in *Developing Fusion Web Applications with Oracle Application Development Framework*.

An MDS Repository can be file-based or database-based. For production environments, you use a database-based repository. You can have more than one MDS Repository for a domain.

A database-based MDS Repository provides the following features that are not supported by a file-based MDS Repository:

- **Efficient query capability:** A database-based MDS Repository is optimized for set-based queries. As a result, it provides better performance on such searches with the database repository.

The MDS Repository query API provides constructs to define the query operation and to specify conditions on metadata objects. These conditions are a set of criteria that restrict the search results to a certain set of attribute types and values, component types, text content, and metadata paths. The API allows multiple conditions to be combined to achieve dynamic recursive composition using OR and AND constructs.

- **Atomic transaction semantics:** A database-based MDS Repository uses the database transaction semantics, which provides rollbacks of failed transactions, such as failed imports or deployments.
- **Versioning:** A database-based MDS Repository maintains versions of the documents in a database-based repository. Versioning allows changes to metadata objects to be stored as separate versions rather than simply overwriting the existing data in the metadata repository. It provides version history, as well as the ability to label versions so that you can access the set of metadata as it was at a given point in time.

- Isolate metadata changes: A database-based MDS Repository has the capability to isolate metadata changes in a running environment and test them for a subset of users before committing them for all users.
- Support for external change detection based on polling: This allows one application to detect changes that another application makes to shared metadata. For example, if you have an application deployed to Managed Servers A and B in a cluster, and you modify the customizations for the application deployed to Managed Server A, the data is written to the database-based repository. The application deployed to Managed Server B uses the updated customizations. This supports high availability (in particular, active/active scenarios.)
- Clustered updates: A database-based MDS Repository allows updates from multiple hosts to the metadata. For a file-based MDS Repository, updates can be made from only one host at a time.

Multiple applications can share metadata by configuring a shared metadata repository. When you do this, changes made by one application to the metadata in this repository are seen by other applications using the shared repository, if you configure external change detection for the applications.

In an MDS Repository, each application, including Oracle Fusion Middleware components, is deployed to its own partition. A **partition** is an independent logical repository within one physical MDS Repository, whether it is database-based or file-based.

For information about deploying applications and associating them with an MDS Repository, see [Chapter 9](#).

Note the following points about patching the MDS Repository:

- An MDS Repository must be registered with a domain before it is patched. Otherwise, the applied patches cannot be rolled back and no additional patches can be applied.
- You can apply patches to the following:
 - The MDS metadata
 - An MDS jar file
 - An MDS shared library
 - An MDS schema in the database-based metadata repository. The patch can include additive changes such as adding a new column or increasing the size of a column. Note that you cannot rollback this type of patch.
 - The MDS database PL/SQL in the database-based metadata repository. The patch can include changes to a PL/SQL package or new PL/SQL packages and procedures.
 - An MDS schema or PL/SQL in the database-based metadata repository that requires a corresponding MDS JAR file patch.

13.3.1.1 Databases Supported by MDS

The MDS Repository supports Oracle databases, as well as non-Oracle databases, including SQL Server, DB2, and MySQL.

For information about the supported databases and the supported versions, as well as using these databases with the MDS Repository, see "Supported Databases for the MDS Schema" in the *Oracle Fusion Middleware System Requirements and Specifications*.

13.3.1.2 Understanding MDS Operations

You can use Fusion Middleware Control or WLST commands to perform most operations on the MDS Repository. However, for some operations that do not have a custom user interface in Fusion Middleware Control or do not have WLST commands, you must use the System MBeans.

The sections that follow describe using Fusion Middleware Control and WLST commands to perform the operations, unless only System MBeans are supported. In that case, the sections describe how to use System MBeans to perform the operation.

You can view information about the repositories, including the partitions and the applications deployed to each partition. You can also perform operations on the partitions, such as purging, deleting, importing metadata, or exporting metadata.

Note the following when you use the MDS operations described in the sections that follow:

- The export operation exports a versioned stripe (either the tip version or based on a label) of metadata documents from an MDS Repository partition to a file system directory or archive. If you export to a directory, the directory must be accessible from the host where the application is running. If you export to an archive, the archive can be located on the system on which you are executing the command.

Because versioning of metadata is not supported for file-based repositories, the tip version (which is also the only version) is exported from a file-based repository.

- The import operation imports metadata documents from a file system directory or archive to a MDS Repository partition. If you exported to a directory, the directory must be accessible from the host where the application is running. If you exported to an archive, the archive can be located on the system on which you are executing the command.

If the target repository is a database-based repository, the metadata documents are imported as new tip versions. If the target repository is a file-based repository, the metadata documents are overwritten.

Note:

- To use the custom WLST MDS commands, you must invoke the WLST script from the Oracle Common home. See [Section 2.4.2](#) for more information.
 - For more information about the custom WLST MDS commands, see "Metadata Services (MDS) Custom WLST Commands" in the *WLST Command Reference for Infrastructure Components*.
-
-

[Table 13–1](#) lists the logical roles needed for each operation. The roles apply whether the operations are performed through the WLST commands, Fusion Middleware Control, or MBeans.

Table 13–1 MDS Operations and Required Roles

Operation	Logical Role
Clear cache	Operator role for application
Clone metadata partition	Admin role for domain
Create metadata label	Admin role for application
Create metadata partition	Admin role for domain

Table 13–1 (Cont.) MDS Operations and Required Roles

Operation	Logical Role
Delete metadata	Admin role for application
Delete metadata label	Admin role for application
Delete metadata partition	Admin role for domain
Deregister metadata database repository	Admin role for domain
Deregister metadata file repository	Admin role for domain
Destroy sandbox	Admin role for application
Export metadata	Monitor role for application
Export sandbox metadata	Monitor role for application
Import MAR	Admin role for application
Import metadata	Admin role for application
Import sandbox metadata	Admin role for application
List metadata label	Monitor role for application
List sandboxes	Monitor role for application
Promote metadata label	Admin role for application
Purge metadata	Admin role for application
Purge metadata labels	Admin role for application
Register metadata database repository	Admin role for domain
Register metadata file repository	Admin role for domain

For information about how these roles map to WebLogic Server roles, see "Mapping of Logical Roles to WebLogic Roles" in *Securing Applications with Oracle Platform Security Services*.

13.3.2 Registering and Deregistering a Database-Based MDS Repository

The following topics describe how to register and deregister a database-based MDS Repository:

- [Registering a Database-Based MDS Repository Using Fusion Middleware Control](#)
- [Deregistering a Database-Based MDS Repository](#)

Note: Note the following if you invoke the following WLST commands or comparable MBeans in a script:

- registerMetadataDBRepository
- deregisterMetadataDBRepository

In this release and previous releases, the commands or MBeans have the following behavior:

1. Starts an Oracle WebLogic Server editing session.
2. Registers or deregisters the repository.
3. Activates the changes.

However, you can start an editing session explicitly. If you do, the automatic activation of the changes are deprecated.

13.3.2.1 Registering a Database-Based MDS Repository

Before you can deploy an application to an MDS Repository, you must register the repository with the Oracle WebLogic Server domain. You can register a database-based MDS Repository using Fusion Middleware Control or WLST, as described in the following topics:

- [Registering a Database-Based MDS Repository Using Fusion Middleware Control](#)
- [Registering a Database-Based MDS Repository Using WLST](#)

13.3.2.1.1 Registering a Database-Based MDS Repository Using Fusion Middleware Control

You create a database-based MDS Repository using RCU, as described in [Section 13.2](#).

To register a database-based MDS Repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services**, then **Metadata Repositories**.

The Metadata Repositories page is displayed, as shown in the following figure:

The screenshot shows the 'Metadata Repositories' page in Fusion Middleware Control. The page title is 'base_domain' and the user is logged in as 'weblogic'. The page was refreshed on May 28, 2013, at 7:58:31 AM PDT. The main content area is titled 'Metadata Repositories' and contains a sub-section 'Database-Based Repositories'. This section has 'Register...' and 'Deregister...' buttons. Below these buttons is a table with the following data:

Repository Name	Database Type	Database Name	Schema Name
mds-owsm	Oracle	orcl.us.oracle.com	DEV6_MDS

Below the table is a scroll bar. Underneath the 'Database-Based Repositories' section is a 'File-Based Repositories' section, which also has 'Register...' and 'Deregister...' buttons. Below this section, it says 'No Repository'.

2. In the Database-Based Repositories section, click **Register**.
The Register Database-Based Metadata Repository page is displayed.
3. In the Database Connection section, enter the following information:
 - For **Database Type**, select the type of database.
 - For **Host Name**, enter the name of the host.
 - For **Port**, enter the port number for the database, for example: 1521.
 - For **Service Name**, enter the service name for the database. The default service name for a database is the global database name, comprising the database name, such as `orcl`, and the domain name. In this case, the service name would be `orcl.domain_name.com`.
 - For **User Name**, enter a user name for the database which is assigned the SYSDBA role, for example: `SYS`.
 - For **Password**, enter the password for the user.
 - For **Role**, select a database role, for example, **SYSDBA**.
4. Click **Query**.

A table is displayed that the metadata repositories in the database, as shown in the following figure:

The screenshot shows the 'Register Database-Based Metadata Repository' page. The 'Database Connection Information' section is filled with the following values:

- Database Type: Oracle
- Host Name: host.example.com
- Port: 1521
- Service Name: orcl.us.oracle.com
- User Name: sys
- Password: (masked)
- Role: SYSDBA

Below the form is a table showing existing metadata repositories:

Metadata Repository	Is Registered?	Schema Name	Version	Status	Modifi
MDS	false	DEV1212_MDS	12.1.2.0.0	VALID	Apr 30
MDS	false	DEV1_MDS	12.1.2.0.0	VALID	Apr 30
MDS	false	DEV11G_MDS	11.1.1.7.0	VALID	Apr 30
MDS	true	DEV6_MDS	12.1.2.0.0	VALID	May 9
MDS	false	DEV3_MDS	12.1.2.0.0	VALID	Apr 30
MDS	false	DEV5_MDS	12.1.2.0.0	VALID	Apr 30
MDS	false	DEV4_MDS	12.1.2.0.0	VALID	Apr 30

5. Select a repository, then enter the following information:
 - For **Repository Name**, enter a name.
 - For **Schema Password**, enter the password you specified when you created the schema.
6. Click **OK**.

The repository is registered with the Oracle WebLogic Server domain and is targeted to the Administration Server. To target the repository to other servers, see [Section 13.3.2.2](#).

In addition, a system data source is created with the name `mds-repository_name`. Global transaction support is disabled for the data source.

13.3.2.1.2 Registering a Database-Based MDS Repository Using WLST To register a database-based MDS Repository using the command line, you use the WLST `registerMetadataDBRepository` command. For example, to register the MDS Repository `mds-repos1`, use the following command:

```
registerMetadataDBRepository (name='mds-repos1', #dbVendor='ORACLE',
###host='hostname', #port='1521', #dbName='ora11', #
###user='username', #password='password', #targetServers='server1')
```

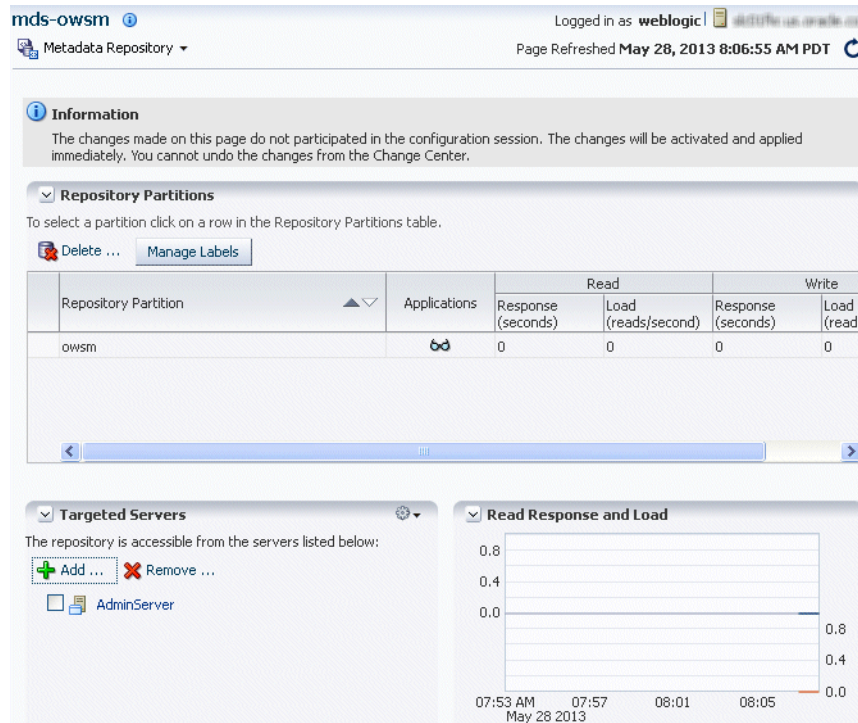
13.3.2.2 Adding or Removing Servers Targeted to the MDS Repository

When you register an MDS Repository using Fusion Middleware Control, the repository is targeted to the Administration Server. You can target the repository to additional servers or remove servers as targets.

To target the MDS Repository to additional servers:

1. From the navigation pane, expand **Metadata Repositories**.
2. Select the repository.

The repository home page is displayed, as shown in the following figure:



3. In the Targeted Servers section, click **Add**.
The Target the Repository dialog box is displayed.
4. Select the server or cluster and click **Target**.
You can expand the cluster to see the servers in the cluster. However, if you select a cluster, the repository is targeted to all servers in the cluster.
5. When the operation completes, click **Close**.
The server is now listed in the Targeted Servers section.

To remove a server as a target for the repository:

1. From the navigation pane, expand **Metadata Repositories**.

2. Select the repository.
The repository home page is displayed.
3. In the Targeted Servers section, select the target server and click **Remove**.
The Untarget the Repository dialog box is displayed.
4. Select the server or cluster and click **Untarget**.
You can expand the cluster to see the servers in the cluster. However, if you select a cluster, the repository will be untargeted from all servers in the cluster.
5. When the operation completes, click **Close**.

13.3.2.3 Deregistering a Database-Based MDS Repository

Deregistration does not result in loss of data stored in the repository. However, any applications using a deregistered repository will not function after the repository is deregistered. You must ensure that no application is using the repository before you deregister it.

You can deregister a database-based MDS Repository using Fusion Middleware Control or WLST, as described in the following topics:

- [Deregistering a Database-Based MDS Repository Using Fusion Middleware Control](#)
- [Deregistering a Database-Based MDS Repository Using WLST](#)

13.3.2.3.1 Deregistering a Database-Based MDS Repository Using Fusion Middleware Control

To deregister an MDS Repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services**, then **Metadata Repositories**.

The Metadata Repositories page is displayed.

Alternatively, you can navigate to the Register Metadata Repositories page by choosing **Administration**, then **Register/Deregister** from the Metadata Repository menu when you are viewing a metadata repository home page.

2. Select the repository from the table.
3. Click **Deregister**.
4. Click **Yes** in the Confirmation dialog box.

13.3.2.3.2 Deregistering a Database-Based MDS Repository Using WLST To deregister a database-based MDS Repository using the command line, you use the WLST `deregisterMetadataDBRepository` command. For example, to deregister the MDS Repository `mds-repos1`, use the following command:

```
deregisterMetadataDBRepository(name='mds-repos1')
```

13.3.3 Registering and Deregistering a File-Based MDS Repository

The following topics describe how to register and deregister a file-based metadata repository:

- [Creating and Registering a File-Based MDS Repository](#)
- [Deregistering a File-Based MDS Repository](#)

Note: Note the following if you invoke the following MBeans in a script:

- registerMetadataFileRepository
- deregisterMetadataFileRepository

In this release and previous releases, the MBeans have the following behavior:

1. Starts an Oracle WebLogic Server editing session.
2. Registers or deregisters the repository.
3. Activates the changes.

However, you can start an editing session explicitly. If you do, the automatic activation of the changes are deprecated.

13.3.3.1 Creating and Registering a File-Based MDS Repository

You can create a file-based MDS Repository and register it with an Oracle WebLogic Server domain using Fusion Middleware Control.

To create and register a file-based repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services**, then **Metadata Repositories**.

The Metadata Repositories page is displayed.

2. In the File-Based Repository section, click **Register**.

The Register Metadata Repository page is displayed.

3. Enter the following information:

- For **Name**, enter a name. For example, enter repos1. The prefix mds- is added to the name and a repository with the name mds-repos1 is registered. If you enter a name that begins with mds-, a repository with the given name is registered.
- For **Directory**, specify the directory. The Administration Server and Managed Servers that run the applications that use this repository must have write access to the directory.

Note the following:

- If the specified path exists on the file system, the metadata file repository is registered; all the subdirectories under this path are automatically loaded as partitions of this file-based repository.
- If the path specified does not exist, a directory with this name is created on the file system during the registration. Because there are no partitions created yet, there are no subdirectories to load.
- If the specified path is invalid and cannot be created for some reason, such as permission denied, an error is displayed and the registration fails.
- If the specified path exists, but as a file not a directory, an error is not displayed and the registration succeeds.

4. Click **OK**.

The repository is created and registered and is displayed on the Metadata Repositories page.

You can now create and delete partitions. Those changes are reflected in the directory on the file system.

You can also create a file-based repository using system MBeans. For information about using the System MBean Browser, see [Section 13.3.5](#).

13.3.3.2 Deregistering a File-Based MDS Repository

You can deregister a file-based MDS Repository using Fusion Middleware Control.

To deregister a file-based repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services**, then **Metadata Repositories**.

The Metadata Repositories page is displayed.

2. In the File-Based Repository section, select the repository and click **Deregister**.

3. Click **OK** in the Confirmation dialog box.

If the file-based repository is valid, it is removed from the repository list. Otherwise, an error is displayed.

You can also deregister a file-based repository using system MBeans. For information about using the System MBean Browser, see [Section 13.3.5](#).

13.3.4 Changing the System Data Source

You can change the system data source to reassociate an application to a new repository. You can change the database or the schema that contains the data source. To do so, you can use Oracle WebLogic Server Administration Console or Fusion Middleware Control. To use Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **JDBC Data Sources**.

The JDBC Data Sources page is displayed.

2. Select the data source you want to change.

The JDBC Data Source page for the selected data source is displayed.

3. Select the Connection Pool tab.

4. To change the database, modify the **Database URL** field. For example:

```
jdbc:oracle:thin:@hostname.domainname.com:1522/orcl
```

5. For **Password**, enter the password for the database.
6. To change the schema, modify the Properties section, changing the value for **user**.
7. If the database is a DB2 database, add the property `sendStreamAsBlob`, with a value of `true`.
8. Click **Apply**.
9. Restart the servers that use this data source.

13.3.5 Using System MBeans to Manage an MDS Repository

Although most procedures in this chapter discuss using Fusion Middleware Control or WLST to manage the MDS Repository, you can also use system MBeans:

1. From the WebLogic Domain menu, choose **System MBean Browser**.

The System MBean Browser page is displayed.

2. In the page's navigation pane, expand **Application Defined MBeans**, then expand **oracle.mds.lcm**. Expand the domain, then **MDSDomainRuntime**, and then select **MDSDomainRuntime**.
3. In the Application Defined MBeans pane, select the Operations tab.
4. Click one of the operations, such as **registerMetadataFileRepository**.
The Operations page is displayed.
5. In the Value column, enter values for the operation.
6. Click **Invoke**.

13.3.6 Viewing Information About an MDS Repository

You can view information about an MDS Repository using Fusion Middleware Control or system MBeans, as described in the following topics:

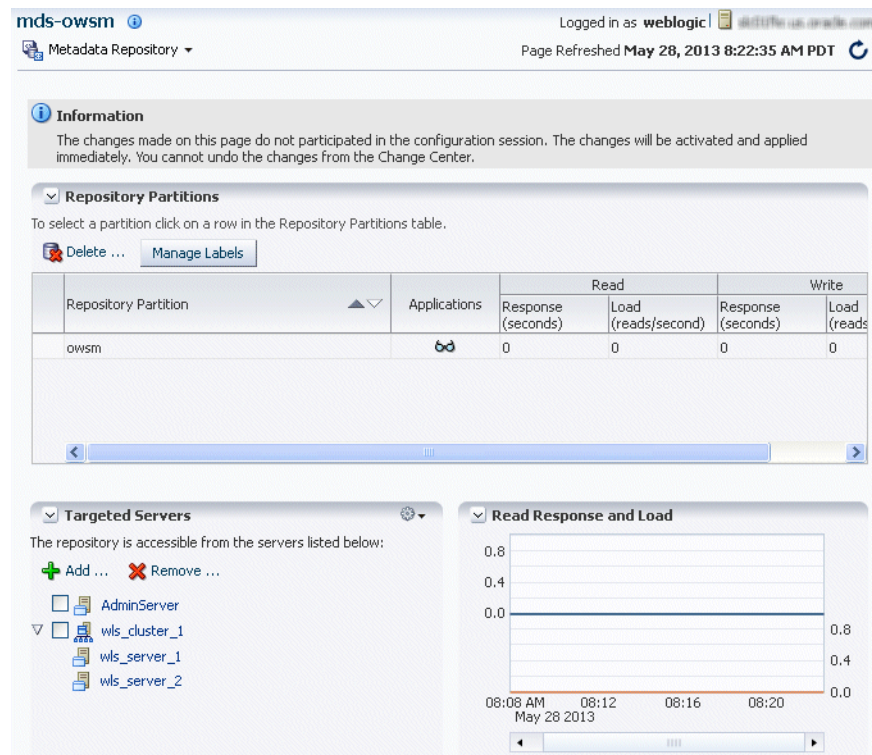
- [Viewing Information About an MDS Repository Using Fusion Middleware Control](#)
- [Viewing Information About an MDS Repository Using System MBeans](#)

13.3.6.1 Viewing Information About an MDS Repository Using Fusion Middleware Control

To view information about an MDS Repository using Fusion Middleware Control:

1. From the navigation pane, expand **Metadata Repositories**.
2. Select the repository.

The following figure shows the home page for an MDS Repository:



3. To see which applications use the repository, click the icon in the Applications column. The Applications using the partition dialog box is displayed, with tabs for Deployed Applications and Referenced by Applications:
 - The Deployed Applications tab shows the list of applications whose metadata is deployed to the repository partition.
 - The Referenced by Applications tab shows the list of applications that refer to the metadata stored in the repository partition.

From this page, you can also:

- Delete partitions, as described in [Section 13.3.10.1](#).
- Delete labels, as described in [Section 13.3.12.5](#).
- Add or remove targeted servers, as described in [Section 13.3.2.2](#).

13.3.6.2 Viewing Information About an MDS Repository Using System MBeans

You can use the System MBean operations `listPartitions`, `listRepositories`, and `listRepositoryDetails` to get a list of partitions in the repository, a list of repositories, and details of the repository registered with the domain:

1. From the WebLogic Domain menu, choose **System MBean Browser**.
The System MBean Browser page is displayed.
2. In the page's navigation pane, expand **Application Defined MBeans**, then expand **oracle.mds.lcm**. Expand the domain, then **MDSDomainRuntime**, and then select **MDSDomainRuntime**.
3. In the Application Defined MBeans pane, select the Operations tab.
4. Click one of the operations, such as `listPartitions`, `listRepositories`, and `listRepositoryDetails`.
The Operations page is displayed.
5. Click **Invoke**.
The information is displayed in the Return Value table.

For information about changing the MDS configuration attributes for an application, see [Section 9.7](#).

13.3.7 Configuring an Application to Use a Different MDS Repository or Partition

When you deploy an application, you can associate it with an MDS Repository. You can subsequently change the MDS Repository or partition to which an application is associated, using WLST or Fusion Middleware Control. For example, a different repository contains different metadata that needs to be used for a particular application.

To associate an application with a new MDS Repository or partition, you can either:

- Redeploy the application, specifying the new repository or partition.
To create a new partition, you can either:
 - Clone the partition to a different repository. Cloning the partition is valid only with a database-based repository with databases of the same type and version. When you clone the partition, you preserve the metadata version history, including any customizations and labels.

[Section 13.3.7.1](#) describes how to clone a partition and how to redeploy the application, specifying the partition that you have cloned.

- Create a new partition, then export the metadata from the current partition and import the metadata into the new partition.

[Section 13.3.7.2](#) describes how to create the partition and export and import data and how to redeploy the application, specifying the new repository or partition.

- Change the system data source. When you change the system data source, you can change the database or the schema in which it is stored.

[Section 13.3.4](#) describes how to change the system data source.

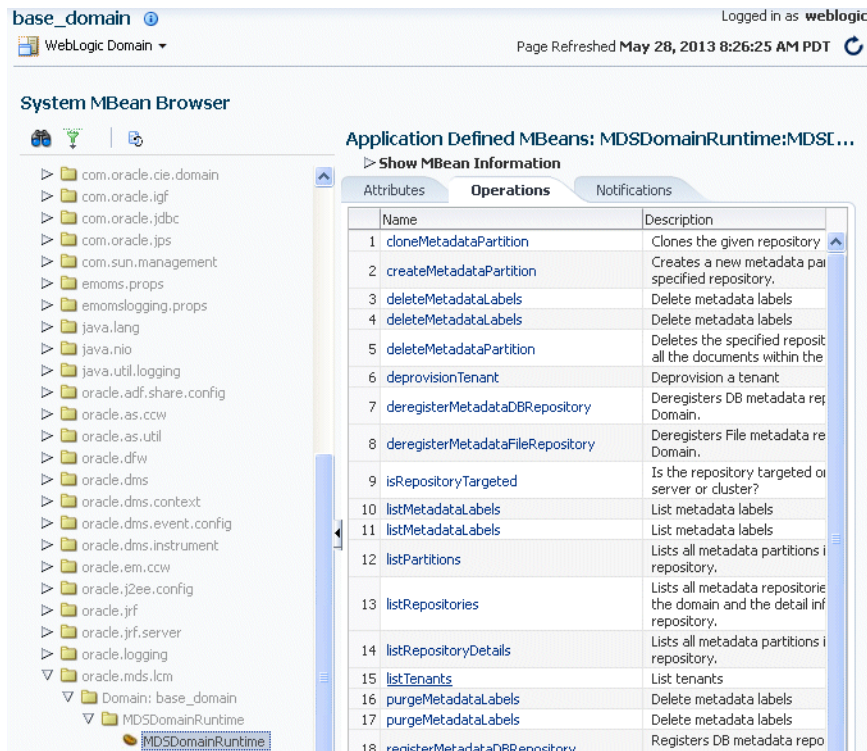
13.3.7.1 Cloning a Partition

You can clone a partition to the same repository or a different repository using the system MBean `cloneMetadataPartition`. Both the original repository and the target repository must be a database-based repository.

To clone the partition, and then redeploy the application to a new repository or to the same repository:

1. Clone the partition, using the `cloneMetadataPartition` operation on the system MBean. The following example clones `partition1` from the old repository to the new repository:
 - a. In Fusion Middleware Control, from the navigation pane, navigate to the Managed Server from which the application is deployed. From the WebLogic Server menu, choose **System MBean Browser**.
The System MBean Browser page is displayed.
 - b. In the System MBean Browser's navigation pane, expand **Application Defined MBeans**, then expand **oracle.mds.lcm**. Expand the domain, and then **MDSDomainRuntime**. Select **MDSDomainRuntime**.
 - c. In the Application Defined MBeans pane, select the Operations tab.

The following figure shows the System MBeans Browser with the Application Defined MBeans pane:



d. Select cloneMetadataPartition.

The Operation: cloneMetadataPartition page is displayed.

e. In the Parameters table, enter the following values:

- For **fromRepository**, enter the name of the metadata repository that contains the metadata partition from which the metadata documents are to be cloned.
- For **fromPartition**, enter the name of the partition from which the metadata documents are to be cloned.
- For **toRepository**, enter the name of the metadata repository to which the metadata documents from the source repository partition are to be cloned.
- For **toPartition**, enter the name of metadata repository partition to be used for the target partition. The name must be unique within the repository. If you do not supply a value for this parameter, the name of the source partition is used for the target partition.

If the toRepository name is the same as the original repository, you must enter a partition name and the name must be unique within the repository.

f. Click Invoke.

g. Verify that the partition has been created by selecting the repository in the navigation pane. The partition is listed in the Partitions table on the Metadata Repository home page.

2. Redeploy the application, as described in Section 9.4.3. When you do so, you specify the new partition and repository in the Application Attributes page:

- a.** To change the repository, click the icon next to the **Repository Name**. In the Metadata Repositories dialog box, select the repository and click **OK**.
- b.** To change the partition, enter the partition name in **Partition Name**.

13.3.7.2 Creating a New Partition and Reassociating the Application to It

You can create a new partition in the same or a different repository by redeploying the application and specifying the new partition. Then, you transfer the metadata to the new partition using WLST.

You can use this procedure to transfer metadata between two different types of repositories (file-based to database-based or from an Oracle Database to another database.)

To create a new partition and reassociate the application to it:

1. Export the metadata from the source partition to a directory on the file system using the WLST `exportMetadata` command:

```
exportMetadata(application='sampleApp',#server='server1',
#####toLocation='/tmp/myrepos/mypartition',#docs='/**')
```

2. Redeploy the application, as described in [Section 9.4.3](#). When you do so, you specify the new partition and repository in the Application Attributes page:
 - a. To change the repository, click the icon next to the **Repository Name**. In the Metadata Repositories dialog box, select the repository and click **OK**.
 - b. To change the partition, enter the partition name in **Partition Name**.
3. Import the metadata from the file system to the new partition using the WLST `importMetadata` command:

```
importMetadata(application='sampleApp',#server='server1',
#####fromLocation='/tmp/myrepos/mypartition',#docs='/**')
```

4. Optionally, deregister the original repository, as described in [Section 13.3.3.2](#) or [Section 13.3.2.3](#).

Alternatively, you can create a new partition using the WLST command `createMetadataPartition`. The partition name must be unique within the repository. If the partition parameter is missing, the name of the source partition is used for the target partition. The following example creates the partition `partition1`:

```
createMetadataPartition(repository='mds-repos1',#partition='partition1')
```

13.3.8 Moving Metadata from a Source System to a Target System

You can transfer the metadata in MDS from one partition to another. As an example, you want to move an application from a test system to a production system. You have a test application that is deployed in a domain in the test system and a production application deployed in a domain in the production system. You want to transfer the customizations from the test system to the production system. To do that, you transfer the metadata from the partition in the test system to a partition in the production system.

To transfer the metadata from one partition to another, you export the metadata from the partition and then import it into the other partition. You can use Fusion Middleware Control or WLST to transfer the metadata, as described in the following topics:

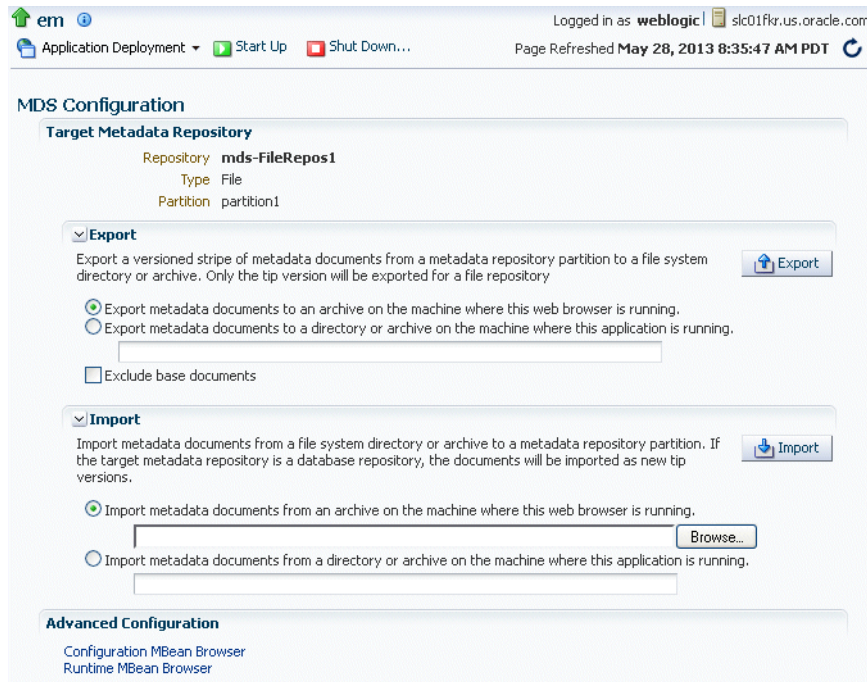
- [Transferring Metadata Using Fusion Middleware Control](#)
- [Transferring Metadata using WLST](#)

13.3.8.1 Transferring Metadata Using Fusion Middleware Control

To use Fusion Middleware Control to transfer metadata:

1. From the navigation pane, expand **Application Deployments**, then select the application.
2. From the Application Deployment menu, choose **MDS Configuration**.

The MDS Configuration page is displayed, as shown in the following figure:



3. In the Export section, select one of the following:
 - **Export metadata documents to an archive on the machine where this web browser is running.**

Click **Export**.

The export operation exports a zip file. Depending on the operating system and browser, a dialog box is displayed that asks you if you want to save or open the file.

- **Export metadata documents to a directory or archive on the machine where this application is running.**

Enter a directory location or archive to which the metadata can be exported.

The target directory or archive file (.jar, .JAR, .zip or .ZIP) to which to transfer the documents selected from the source partition. If you export to a directory, the directory must be a local or network directory or file where the application is physically deployed. If you export to an archive, the archive can be located on a local or network directory or file where the application is physically deployed, or on the system on which you are executing the command.

If the location does not exist in the file system, a directory is created except that when the names ends with .jar, .JAR, .zip or .ZIP, an archive file is created. If the archive file already exists, the exportMetadata operation overwrites the file.

Click **Export**. Then, in the Confirmation dialog box, click **Close**.

If you check **Exclude base documents**, this operation exports only the customizations, not the base documents. See [Section 13.3.1](#) for information about base documents and customizations.

4. If the target application is on a different system, copy the exported metadata to that system.
5. From the navigation pane for the target system, expand **Application Deployments**, then select the application.
6. From the Application Deployment menu, choose **MDS Configuration**.
The MDS Configuration page is displayed
7. In the Import section, select one of the following:
 - **Import metadata documents from an archive on the machine where this web browser is running.**
 - **Import metadata documents from a directory or archive on the machine where this application is running.**

Enter the location of the directory or archive that contains the exported metadata. If you specify a directory, include the subdirectory with the partition name in the specification. The directory or archive file must be a local or network directory or file where the application is physically deployed.

8. Click **Import**.
9. In the Confirmation dialog box, click **Close**.

13.3.8.2 Transferring Metadata using WLST

To use WLST to transfer metadata:

1. Export the metadata from the original partition using the `exportMetadata` command:

```
exportMetadata(application='sampleApp',#server='server1',
#####toLocation='/tmp/myrepos/mypartition',#docs='/**')
```

This command exports a versioned stripe of the metadata documents from the metadata partition to a file system directory. Only customization classes declared in the `cust-config` element of `adf-config.xml` are exported. If there is no `cust-config` element declared in `adf-config.xml`, all customization classes are exported.

To export all customizations, use the option `restrictCustTo=""`.

2. If the production application is on a different system, copy the exported metadata to that system.
3. Import the metadata to the other partition using the WLST `importMetadata` command:

```
importMetadata(application='sampleApp',#server='server1',
#####fromLocation='/tmp/myrepos/mypartiton',#docs='/**')
```

The value of the `fromLocation` parameter must be on the same system that is running WLST or on a mapped network drive or directory mount. You cannot use direct network references such as `\\mymachine\repositories\`.

Only customization classes declared in the cust-config element of adf-config.xml are imported. If there is no cust-config element declared in adf-config.xml, all customization classes are imported.

To import all customizations, use the option restrictCustTo="%".

13.3.9 Moving from a File-Based Repository to a Database-Based Repository

You can move from a file-based repository to a database-based repository. (You cannot move from a database-based repository to a file-based repository.)

To minimize downtime, take the following steps to move an application's metadata from a file-based repository to a database-based repository:

1. Use RCU to create schemas in the new repository, as described in [Section 13.2](#).
2. Create a new partition using the WLST command `createMetadataPartition` with same name as source partition:

```
createMetadataPartition(repository='mds-repos1',#partition='partition1')
```

3. Export the metadata from the source partition to a directory on the file system:

```
exportMetadata(application='sampleApp',#server='server1',
#####toLocation='/tmp/myrepos/partition1',#docs='/**')
```

4. Import the metadata from the file system to the new partition:

```
importMetadata(application='sampleApp',#server='server1',
#####fromLocation='/tmp/myrepos/partition1',#docs='/**')
```

5. Redeploy the application, as described in [Section 9.4.3](#). When you do so, you specify the new partition and repository in the Application Attributes page:
 - a. To change the repository, click the icon next to the **Repository Name**. In the Metadata Repositories dialog box, select the repository and click **OK**.
 - b. To change the partition, enter the partition name in **Partition Name**.
6. Deregister the file-based repository, as described in [Section 13.3.3.2](#).

13.3.10 Deleting a Metadata Partition from a Repository

You can delete metadata partitions if there are no applications either deployed to the partition or referring to the partition. You may want to delete a metadata partition from the repository in the following circumstances:

- When you undeploy an application. Oracle Fusion Middleware leaves the metadata partition because you may still want the metadata, such as user customizations, in the partition. If you do not need the metadata, you can delete the partition.
- When you have transferred metadata from one partition to another and configured the application to use the new partition.
- When you have cloned a partition and configured the application to use the new partition.

Note that deleting a partition deletes all the data contained in the partition.

You can delete a metadata partition using WLST or Fusion Middleware Control, as described in the following topics:

- [Deleting a Metadata Partition Using Fusion Middleware Control](#)

- [Deleting a Metadata Partition Using WLST](#)

13.3.10.1 Deleting a Metadata Partition Using Fusion Middleware Control

To delete a metadata partition from a repository partition using Fusion Middleware Control:

1. From the navigation pane, expand **Metadata Repositories**.
2. Select the repository.
The repository home page is displayed.
3. In the Repository Partitions section, select the partition and click **Delete**.
4. In the confirmation dialog box, click **OK**.

13.3.10.2 Deleting a Metadata Partition Using WLST

To delete a metadata partition from a repository, you can use the WLST command `deleteMetadataPartition`. For example, to delete the metadata partition from the file-based repository `mds-repos1`, use the following command:

```
deleteMetadataPartition(repository='mds-repos1',#partition='partition1')
```

13.3.11 Purging Metadata Version History

For database-based MDS Repositories, you can purge the metadata version history from a partition. (File-based MDS Repositories do not maintain version history.) This operation purges version history of unlabeled documents from the application's repository partition. The tip version (the latest version) is not purged, even if it is unlabeled.

To purge metadata labels, you use the `purgeMetadataLabels` command, as described in [Section 13.3.12.4](#). Then, you can purge the metadata version history.

Consider purging metadata version history on a regular basis as part of MDS Repository maintenance, when you suspect that the database is running out of space or performance is becoming slower. This operation may be performance intensive, so plan to do it in a maintenance window or when the system is not busy.

For specific recommendations for particular types of applications, see the documentation for a particular component.

You can purge metadata version history using WLST or Fusion Middleware Control, as described in the following topics:

- [Purging Metadata Version History Using Fusion Middleware Control](#)
- [Purging Metadata Version History Using WLST](#)
- [Enabling Auto-Purge](#)

13.3.11.1 Purging Metadata Version History Using Fusion Middleware Control

To use Fusion Middleware Control to purge the metadata version history:

1. From the navigation pane, expand **Application Deployments**, then select the application.
2. From the Application Deployment menu, choose **MDS Configuration**.
The MDS Configuration page is displayed.

3. In the Purge section, in the **Purge all unlabeled past versions older than** field, enter a number and select the unit of time. For example, enter **3** and select **months**.
4. Click **Purge**.
5. In the Confirmation dialog box, click **Close**.

13.3.11.2 Purging Metadata Version History Using WLST

To use WLST to purge metadata version history, use the `purgeMetadata` command. You specify the documents to be purged by using the `olderThan` parameter, specifying the number of seconds. The following example purges all documents older than 100 seconds:

```
purgeMetadata(application='sampleApp',#server='server1',#olderThan=100)
```

13.3.11.3 Enabling Auto-Purge

You can enable automatic purging using the MDSAppConfig MBean:

1. From the WebLogic Domain menu, choose **System MBean Browser**.
The System MBean Browser page is displayed.
2. Expand **Application Defined MBeans**, then **oracle.adf.share.config**, then **Server: name**, then **Application: name**, then **ADFConfig**, then **ADFConfig**, and **ADFConfig**.
3. Select **MDSAppConfig**.
The Application Defined MBeans page is displayed.
4. For **AutoPurgeTimeToLive**, enter a value, in seconds.
5. Navigate up to ADFConfig (the parent of MDSAppConfig) and select it.
6. In the Operations tab, click **Save**.

13.3.12 Managing Metadata Labels in the MDS Repository

A **metadata label** is a means of selecting a particular version of each object from a metadata repository partition. Conceptually, it is a collection of document versions, one version per document, representing a *horizontal stripe* through the various document versions. This stripe comprises the document versions which were the tip versions (latest versions) at the time the label was created.

You can use a label to view the metadata as it was at the point in time when the label was created. You can use the WLST commands to support logical backup and recovery of an application's metadata contained in the partition.

Labels are supported only in database-based repositories.

Document versions belonging to a label are not deleted by automatic purging, unless the label is explicitly deleted. In this way, creating a label guarantees that a view of the metadata as it was at the time the label was created remains available until the label is deleted.

When an application that contains a MAR is deployed, a label with the prefix `postDeployLabel_` is created. For example: `postDeployLabel_mdappdb_mdappdb.mar_2556916398`.

Each time you patch the MAR, a new deployment label is created, but the previous deployment label is not deleted. Similarly, when you undeploy an application that

contains a MAR, the application is undeployed, but the label remains in the metadata repository partition.

If you delete a deployment label, when the application is restarted, the MAR is automatically redeployed, and the deployment label is also re-created.

The following topics describe how to manage labels:

- [Creating Metadata Labels](#)
- [Listing Metadata Labels](#)
- [Promoting Metadata Labels](#)
- [Purging Metadata Labels](#)
- [Deleting Metadata Labels](#)

13.3.12.1 Creating Metadata Labels

To create a label for a particular version of objects in a partition in an MDS Repository, you use the WLST command `createMetadataLabel`. For example, to create a label named `prod1` for the application `my_mds_app`, use the following command:

```
createMetadataLabel(application='my_mds_app',#server='server1',#name='prod1')
Executing#operation:#createMetadataLabel.
```

```
Created#metadata#label#"prod1".
```

If the application has more than one version, you must use the `applicationVersion` parameter to specify the version.

13.3.12.2 Listing Metadata Labels

You can list the metadata labels for a particular application. To do so, use the WLST command `listMetadataLabel`. For example, to list the labels for the application `my_mds_app`, use the following command:

```
listMetadataLabels(application='my_mds_app',#server='server1')
Executing#operation:#listMetadataLabels.
```

```
Database#Repository#partition#contains#the#following#labels:
prod1
prod2
postDeployLabel_mdsappdb_mdsappdb.mar_2556916398
```

If the application has more than one version, you must use the `applicationVersion` parameter to specify the version.

13.3.12.3 Promoting Metadata Labels

You can promote documents associated with a metadata label so that they become the latest version. That is, you can promote them to the tip. Promote a label if you want to roll back to an earlier version of all of the documents captured by the label.

To promote a label to the tip, use the WLST command `promoteMetadataLabel`. For example to promote the label `prod1`, use the following command:

```
promoteMetadataLabel(application='my_mds_app',#server='server1',#name='prod1')
Location#changed#to#domainRuntime#tree.#This#is#a#read-only#tree#with#DomainMBean#
as#the#root.#
For#more#help,#use#help(domainRuntime)
```

```
Executing#operation:#promoteMetadataLabel.
```

Promoted#metadata#label#"prod1"#to#tip.

If the application has more than one version, you must use the applicationVersion parameter to specify the version.

13.3.12.4 Purging Metadata Labels

You can purge metadata labels that match the given name pattern or age, allowing you to purge labels that are no longer in use. This reduces the size of the database, improving performance. You must delete the labels associated with unused metadata documents before you can purge the documents and revision history from the repository.

You may want to delete a label for older applications that were undeployed, but the labels were not deleted. Each time you patch the MAR, a new label is created, but the previous label is not deleted.

You can use Fusion Middleware Control or WLST to purge metadata labels, as described in the following topics:

- [Purging Metadata Labels Using Fusion Middleware Control](#)
- [Purging Metadata Labels Using WLST](#)

13.3.12.4.1 Purging Metadata Labels Using Fusion Middleware Control To purge metadata labels using Fusion Middleware Control:

1. Expand **Metadata Repositories**.
2. Select the repository.
The repository home page is displayed.
3. Select a partition and click **Manage Labels**.

The Manage Labels page is displayed, as shown in the following figure:



By default, the table lists all metadata labels created in the selected partition that are more than one year old and that are not deployed or associated with a sandbox.

4. To search for a particular label or labels, you can:
 - For **Label Name**, select an operator and enter the filter criteria. The characters are case sensitive. You can use the following wildcards:
 - Percent (%): Matches any number of characters
 - Underscore (_): Matches a single character
 - Backslash (\): Used as an escape character for the wildcards

For example, the string `postDeployLabel%` returns any label beginning with `postDeployLabel`. As a result, it displays labels associated with a deployed MAR.
 - For **Age**, enter a number, such as 2. (The only operator available is Older Than.)
 - For **Age (units)**, select a unit, such as Hours, Days, Weeks, Months, Years. The only operator available is Equals.
5. Click **Search**.
6. By default, labels associated with sandboxes and deployed applications are not shown. To display those labels, select **Sandboxes** or **Deployment** or both. Note the following:
 - You cannot delete a label associated with a sandbox.
 - If you select **Deployment**, the labels that are associated with MAR deployments are displayed.
7. Select the label and click **Delete Selected**.
8. In the confirmation box, click **OK**.

If you want to purge all unused labels, for a particular deployed application:

1. Select **Deployment**.
2. Filter by name, using the string `postDeployLabel_application_name%`.
3. Select all but the latest (which is in use) to delete. (The most recent label---the one that is currently being used---is listed first.)
4. Click **Delete Selected**.

13.3.12.4.2 Purging Metadata Labels Using WLST You can purge metadata labels that match the given pattern or age, using the WLST command `purgeMetadataLabels`. The command purges the labels that match the criteria specified, but it does not delete the metadata documents that were specified by the labels.

For example, to purge all metadata labels that match the specified `namePattern` and that are older than 30 minutes:

```
purgeMetadataLabels(repository='mds-myRepos', #partition='partition1',
#####namePattern='prod*', #olderThanInMin='30')
Location#changed#to#domainRuntime#tree. #This#is#a#read-only#tree#with#DomainMBean#
as#the#root.#
For#more#help, #use#help(domainRuntime)
```

```
Executing#operation:#purgeMetadataLabels.
```

```
The#following#metadata#labels#were#purged:#  
repository=mds-soa,partition=partition1,namePattern=prod*,olderThanInMin=30:#
```

13.3.12.5 Deleting Metadata Labels

To delete a specified metadata label, you use the WLST command `deleteMetadataLabel`. For example, to delete a label named `prod1` for the application `my_mds_app`, use the following command:

```
deleteMetadataLabel(application='my_mds_app',#server='server1',#name='prod1')
```

If the application has more than one version, you must use the `applicationVersion` parameter to specify the version.

To find the labels associated with an application, use the `listMetadataLabels` command, as described in [Section 13.3.12.2](#).

13.4 Managing Metadata Repository Schemas

The following topics describe how to manage the metadata repository schemas:

- [Changing Metadata Repository Schema Passwords](#)
- [Changing the Character Set of the Metadata Repository](#)

13.4.1 Changing Metadata Repository Schema Passwords

The schema passwords are stored in the database.

For example, to change the password of the schema `OFM_MDS`:

1. Connect to the database using `SQL*Plus`. Connect as a user with `SYSDBA` privileges.
2. Issue the following command:

```
SQL>#ALTER#USER#schema#IDENTIFIED#BY#new_password;
```

For example, to change the `OFM_MDS` password to `abc123`:

```
SQL>#ALTER#USER#OFM_MDS#IDENTIFIED#BY#abc123;
```

3. If you change the MDS Repository schema password, you must change the password for the corresponding MDS Repository data source, using Oracle WebLogic Server Administration Console:
 - a. From Domain Structure, expand **Services**, then **Data Sources**.
 - b. Click the data source that is related to the MDS Repository.
 - c. Click the Configuration tab, then the Connection Pool tab.
 - d. For **Password**, enter the new password.
 - e. Click **Save**.
 - f. Restart the Managed Servers that consume the data source.

13.4.2 Changing the Character Set of the Metadata Repository

For information about changing the character set of metadata repository that is stored in an Oracle Database, see *Oracle Database Globalization Support Guide*:

<http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>

Oracle recommends using Unicode for all new system deployments. Deploying your systems in Unicode offers many advantages in usability, compatibility, and extensibility. Oracle Database enables you to deploy high-performing systems faster and more easily while utilizing the advantages of Unicode. Even if you do not need to support multilingual data today, nor have any requirement for Unicode, it is still likely to be the best choice for a new system in the long run and ultimately saves time and money and gives you competitive advantages in the long term.

When storing the metadata in a SQL Server database, if the character set being considered for your locale is not case neutral, the case-sensitive collation must be selected during the creation of the database instance. Unicode support is the default when creating the MDS schema for SQL Server using RCU. You may overwrite this default to use non-unicode schema if that meets your requirements.

13.5 Purging Data

When the amount of data in Oracle Fusion Middleware databases grows very large, maintaining the databases can become difficult and can affect performance. In some cases, Oracle Fusion Middleware automatically purges data. In other cases, Oracle Fusion Middleware provides methods to manage growth, including scripts to purge data that can accumulate over time and that can affect performance.

Many of the Oracle Fusion Middleware components provide scripts written as PL/SQL procedures to purge the data. The scripts are located in:

`ORACLE_HOME/common/sql/component-name_purge_purgetype.sql`

Table 13–2 provides pointers to information about purging data for Oracle Fusion Middleware components.

Table 13–2 Purging Data Documentation

Component	Description
Oracle Application Development Framework	See "Cleaning Up Temporary Storage Tables" in <i>Developing Fusion Web Applications with Oracle Application Development Framework</i> .
Oracle Application Development Framework Business Components	Use the following script to purge rows in the database used by Oracle ADF Business Components to store user session state and temporary persistent collections: <code>ORACLE_COMMON_HOME/common/sql/adfbc_purge_statesnapshots.sql</code> The PS_TXN table is automatically purged.
Oracle WebLogic Server: Oracle Infrastructure Web Services	Use the following script to purge data if WS-RM uses a database store: <code>ORACLE_COMMON_HOME/common/sql/ows_purge_wsrmsg.sql</code>
Oracle WebLogic Server: JAXWS Web Services	Clean up the Web service persistence store, as described in "Cleaning Up Web Service Persistence" in <i>Developing JAX-WS Web Services for Oracle WebLogic Server</i> . Use the defaultMaximumObjectLifetime field of the WebServicePersistenceMBean to set the maximum lifetime of the objects. See "Understanding WebLogic Server MBeans" in <i>Developing Custom Management Utilities Using JMX for Oracle WebLogic Server</i> .

Table 13–2 (Cont.) Purging Data Documentation

Component	Description
Oracle WebLogic Server: Stateful EJBs	No configuration required. Automatically purges data.
Oracle WebLogic Server: JMS	See "Configuring Basic JMS System Resources" and "Managing JMS Messages" in <i>Administering JMS Resources for Oracle WebLogic Server</i> . Also see "Tuning WebLogic JMS" in <i>Tuning Performance of Oracle WebLogic Server</i> .
Oracle WebLogic Server: Session persistence for JDBC or file-based data sources	No configuration required. Automatically purges data.
MDS Repository	See Section 13.3.11 for information on automatically and manually purging data.
Oracle Web Services Manager	No configuration required. Automatically purges data.

In certain circumstances, you can consider using Oracle Scheduler to automate the running of the scripts. For example, you may want to set up a scheduled job to purge the last 14 days for completed instances.

Oracle Scheduler, an enterprise job scheduler, is part of Oracle Database. Oracle Scheduler is implemented by the procedures and functions in the DBMS_SCHEDULER PL/SQL package.

For information about Oracle Scheduler, see "Oracle Scheduler Concepts" and "Creating, Running, and Managing Jobs" in the *Oracle Database Administrator's Guide*.

13.5.1 Purging Oracle Infrastructure Web Services Data

Use the following script to purge data if WS-RM uses a database store:

```
ORACLE_COMMON_HOME/common/sql/ows_purge_wsrmsg.sql
```

Changing Network Configurations

This chapter provides procedures for changing the network configuration, such as the host name, domain name, or IP address, of an Oracle Fusion Middleware host and the Oracle database that Oracle Fusion Middleware uses. It also includes information about using the IPv6 protocol with Oracle Fusion Middleware.

It contains the following sections:

- [Changing the Network Configuration of Oracle Fusion Middleware](#)
- [Changing the Network Configuration of a Database](#)
- [Moving Between On-Network and Off-Network](#)
- [Changing Between a Static IP Address and DHCP](#)
- [Using IPv6](#)

14.1 Changing the Network Configuration of Oracle Fusion Middleware

This section describes how to change the host name, domain name, IP address, or any combination of these, of a host that contains the following installation types:

- Oracle WebLogic Server and Java components. When you change the host name, domain name, or IP address of Oracle WebLogic Server, you also automatically change the information for Java components that are deployed to Oracle WebLogic Server.
- Oracle HTTP Server. You can change the host name or the IP address.

You cannot use these procedures to change the network configuration of system components other than Oracle HTTP Server. For information about moving from one environment to another, see [Chapter 19](#).

For information about moving your environment to a different system, see [Chapter 19](#).

The following topics describe how to change the host name, domain name, or IP address:

- [Changing the Network Configuration of an Administration Server](#)
- [Changing the Network Configuration of a Managed Server](#)
- [Changing the Network Configuration of Oracle HTTP Server](#)

14.1.1 Changing the Network Configuration of an Administration Server

You can change the network configuration of an Administration Server using WLST commands:

1. Stop the Administration Server.
2. For the Administration Server, set the machine with the new host name, using the following WLST command, in offline mode:

```
wls:/offline>#readDomain('DOMAIN_HOME')
wls:/offline/sampledomain>#cd#('/Machine/newhostname')
wls:/offline/sampledomain>#machine#=#cmo
wls:/offline/sampledomain>#cd#('/Server/AdminServer')
wls:/offline/sampledomain>#set('Machine',#machine)
wls:/offline/sampledomain>#updateDomain()
wls:/offline/sampledomain>#exit()
```

3. Set the listen port for the Administration Server:

```
wls:/offline/sampledomain>#readDomain('DOMAIN_HOME')
wls:/offline/sampledomain>#cd#('/Server/AdminServer')
wls:/offline/sampledomain>#cmo.setListenPort(8001)
wls:/offline/sampledomain>#updateDomain()
wls:/offline/sampledomain>#exit()
```

4. Start the Administration Server.

14.1.2 Changing the Network Configuration of a Managed Server

You can change the network configuration of a Managed Server using the Oracle WebLogic Server Administration Console.

To change the host name, domain name, or IP address of a Managed Server:

1. Display the Administration Console, as described in [Section 2.3.1](#).
2. In the Change Center, click **Lock & Edit**.
3. Create a machine, which is a logical representation of the computer that hosts one or more WebLogic Servers, and point it to the new host. (From the Home page, select **Machines**. Then, click **New**.) Follow the directions in the Administration Console help.

You must disable Host Name Verification on Administration Servers that access Node Manager, as described in "Using Hostname Verification" in *Administering Security for Oracle WebLogic Server*.

4. Restart the Administration Server.
5. Change the Managed Server configuration to point to the new machine:
 - a. From the left pane of the Console, expand **Environment** and then **Servers**. Then, select the name of the server.
 - b. Select the **Configuration** tab, then the **General** tab. In the **Machine** field, select the machine to which you want to assign the server.
 - c. Change **Listen Address** to the new host.
Click **Save**.
6. Start the Managed Server. You can use the Oracle WebLogic Server Administration Console, WLST, or the following command:

```
DOMAIN_NAME/bin/startManagedWeblogic.sh#managed_server_name#
#####admin_url#
```

The Managed Server connects to the Administration Server and updates its configuration changes.

14.1.3 Changing the Network Configuration of Oracle HTTP Server

To change the network configuration of Oracle HTTP Server in a WebLogic domain or a standalone domain:

1. Perform a backup of your environment before you start this procedure. See [Chapter 16](#).
2. Update your operating system with the new host name, domain name, IP address, or any combination of these. Consult your operating system documentation for information on how to perform the following steps.
 - a. Make the updates to your operating system to properly change the host name, domain name, or IP address.
 - b. Restart the host, if necessary for your operating system.
 - c. Verify that you can ping the host from another host in your network. Be sure to ping using the new host name to ensure that everything is resolving properly.

3. Stop Node Manager as described in [Section 4.2.2](#).

4. Stop Oracle HTTP Server. For example:

```
./stopComponent.sh#ohs1
```

5. Change to the following directory:

```
DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_component_name
```

6. For each configuration file (files ending in .conf):

- Search for the old canonical host name (for example oldhost.example.com) and replace it with the new canonical host name (for example newhost.example.com).
- Search for the old short host name (for example oldhost) and replace it with the new short host name (for example newhost).
- Search for the old IP address and replace it with the new IP address.

7. Restart Node Manager:

```
DOMAIN_HOME/bin/startNodeManager.sh
```

8. Restart Oracle HTTP Server:

```
./startComponent.sh#ohs1
```

14.2 Changing the Network Configuration of a Database

This section describes how to change the host name, domain name, or IP address of a host that contains a database that contains the metadata for Oracle Fusion Middleware components:

The following tasks describe the procedure:

- [Task 1, "Stop All Oracle Fusion Middleware Components"](#)
- [Task 2, "Shut Down the Database"](#)
- [Task 3, "Change the Network Configuration"](#)
- [Task 4, "Change References to the Network Configuration"](#)

- [Task 5, "Start the Database"](#)
- [Task 6, "Change the System Data Source"](#)
- [Task 7, "Restart Your Environment"](#)

Task 1 Stop All Oracle Fusion Middleware Components

Stop all components that use the database, even if they are on other hosts. Stop the Administration Server, the Managed Servers, and all components, as described in [Chapter 4](#).

Task 2 Shut Down the Database

Prepare your host for the change by stopping the database:

1. Set the ORACLE_HOME and ORACLE_SID environment variables.
2. Shut down the listener and database:

```
lsnrctl#stop

sqlplus#/nolog
SQL>#connect#SYS#as#SYSDBA
SQL>#shutdown
SQL>#quit
```

3. Verify that all Oracle Fusion Middleware processes have stopped.
4. To ensure that Oracle Fusion Middleware processes do not start automatically after a restart of the host, disable any automated startup scripts you may have set up, such as `/etc/init.d` scripts.

Task 3 Change the Network Configuration

If you are changing the host name, domain name, or IP address, update your operating system with the new names or IP address, restart the host, and verify that the host is functioning properly on your network. Consult your operating system documentation for information on how to perform the following steps:

1. Make the updates to your operating system to properly change the host name, domain name or IP address.
2. Restart the host, if required by your operating system.
3. Verify that you can ping the host from another host in your network. Be sure to ping using the new host name, domain name, or IP address to ensure that everything is resolving properly.

Task 4 Change References to the Network Configuration

You must modify files that contain the host name, domain name, or IP address, depending on the components that you are using. The following lists some of the files that you may need to modify to change references to the new host name, domain name or IP address:

- `tnsnames.ora`, which is located in:
`ORACLE_HOME/network/admin/tnsnames.ora`
- `listener.ora`, which is located in:
`(UNIX) #ORACLE_HOME/network/admin/listener.ora`
`(Windows) #ORACLE_HOME\network\admin\listener.ora`

- For Oracle HTTP Server, edit the `httpd.conf` file, making the following changes:
 - Update the `Listen` directive with the new host name or IP address and port (if the production environment Oracle HTTP Server is using a different port).
 - Update the `VirtualHost` directive, if the host name, IP address, or port number is defined, with the new values for the production environment.
 - Update any other nondefault directives that were configured at the test environment and have topological (host name, IP address, port number) or other machine-specific information.
- For Oracle HTTP Server, the `PlsqlDatabaseConnectionString` in the `dads.conf` file
- For Oracle HTTP Server, if you use `mod_oradav`, the `ORACONNECTSN` parameter in the `mod_oradav.conf` file
- For Oracle HTTP Server, if you use `mod_plsql`, the `PlsqlDatabaseConnectionString` attribute in the `dads.conf` file
- For Oracle HTTP Server, if you use `mod_wl_ohs`, update the `mod_wl_ohs.conf` file
 - Update the `WebLogicHost`, `WebLogicPort`, or `WebLogicCluster` directives with the host name, IP address, and port number.

This is not an exhaustive list. See [Chapter 19](#) for additional information about files used by components. That chapter describes how to move components, including a database, from a test to a production system, in effect changing the host name.

Task 5 Start the Database

Start the database:

1. Log in to the host as the user that installed the database.
2. Set the `ORACLE_HOME` and `ORACLE_SID` environment variables.
3. On UNIX systems, set the `LD_LIBRARY_PATH`, `LD_LIBRARY_PATH_64`, `LIB_PATH`, or `SHLIB_PATH` environment variables to the proper values. The actual environment variables and values that you must set depend on the type of your UNIX operating system.
4. Start the database and listener:

```
sqlplus#/nolog
SQL>#connect#SYS#as#SYSDBA
SQL>#startup
SQL>#quit
```

```
lsnrctl#start
```

Task 6 Change the System Data Source

Change the system data source to use the new host name, domain name, or IP address for the database, as described in [Section 13.3.4](#).

Task 7 Restart Your Environment

Start the components that use the database:

1. Start all components that use the database, even if they are on other hosts. Start the Administration Server, the Managed Servers, and all components, as described in [Chapter 4](#).
2. If you disabled any processes from automatically starting Oracle Fusion Middleware at the beginning of this procedure, enable them.

14.3 Moving Between On-Network and Off-Network

This section describes how to move an Oracle Fusion Middleware host on and off the network. The following assumptions and restrictions apply:

- The host must contain an instance that does not use an Infrastructure, or both the middle-tier instance and Infrastructure must be on the same host.
- DHCP must be used in loopback mode. Refer to the *Oracle Fusion Middleware System Requirements and Specifications* document for more information.
- Only IP address change is supported; the host name must remain unchanged.
- Hosts in DHCP mode should not use the default host name (`localhost.localdomain`). The hosts should be configured to use a standard host name and the loopback IP should resolve to that host name.
- A loopback adapter is required for all off-network installations (DHCP or static IP). Refer to the *Planning an Installation of Oracle Fusion Middleware* for more information.

14.3.1 Moving from Off-Network to On-Network (Static IP Address)

This procedure assumes you have installed Oracle Fusion Middleware on a host that is off the network, using a standard host name (not `localhost`), and would like to move on to the network and use a static IP address. The IP address may be the default loopback IP, or any standard IP address.

To move on to the network, you can simply connect the host to the network. No updates to Oracle Fusion Middleware are required.

14.3.2 Moving from Off-Network to On-Network (DHCP)

This procedure assumes you have installed on a host that is off the network, using a standard host name (not `localhost`), and would like to move on to the network and use DHCP. The IP address of the host can be any static IP address or loopback IP address, and should be configured to the host name.

To move on to the network:

1. Connect the host to the network using DHCP.
2. Configure the host name to the loopback IP address only.

14.3.3 Moving from On-Network to Off-Network (Static IP Address)

Follow this procedure if your host is on the network, using a static IP address, and you would like to move it off the network:

1. Configure the `/etc/hosts` file so the IP address and host name can be resolved locally.
2. Take the host off the network.

There is no need to perform any steps to change the host name or IP address.

14.4 Changing Between a Static IP Address and DHCP

This section describes how to change between a static IP address and DHCP. The following assumptions and restrictions apply:

- The host must contain all Oracle Fusion Middleware components, including Identity Management components, and any database associated with those components. That is, the entire Oracle Fusion Middleware environment must be on the host.
- DHCP must be used in loopback mode. Refer to *Planning an Installation of Oracle Fusion Middleware* for more information.
- Only IP address change is supported; the host name must remain unchanged.
- Hosts in DHCP mode should not use the default host name (`localhost.localdomain`). The hosts should be configured to use a standard host name and the loopback IP should resolve to that host name.

14.4.1 Changing from a Static IP Address to DHCP

To change a host from a static IP address to DHCP:

1. Configure the host to have a host name associated with the loopback IP address before you convert the host to DHCP.
2. Convert the host to DHCP. There is no need to update Oracle Fusion Middleware.

14.4.2 Changing from DHCP to a Static IP Address

To change a host from DHCP to a static IP address:

1. Configure the host to use a static IP address.
2. There is no need to update Oracle Fusion Middleware.

14.5 Using IPv6

Oracle Fusion Middleware supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6.) Among other features, IPv6 supports a larger address space (128 bits) than IPv4 (32 bits), providing an exponential increase in the number of computers that can be addressable on the Web.

An IPv6 address is expressed as 8 groups of 4 hexadecimal digits. For example:

```
2001:0db8:85a3:08d3:1319:8a2e:0370:7334
```

For information about the support for IPv6 by Oracle Fusion Middleware components, see *Oracle Fusion Middleware System Requirements and Specifications*.

The following topics provide more information about configuring Oracle Fusion Middleware components for IPv6:

- [Configuring Oracle HTTP Server for IPv6](#)

14.5.1 Configuring Oracle HTTP Server for IPv6

To configure Oracle HTTP Server to communicate using IPv6, you modify configuration files in the following directory:

```
(UNIX) #DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_name
(Windows) #DOMAIN_HOME\config\fmwconfig\components\OHS\ohs_name
```

For example, to configure Oracle HTTP Server to communicate with Oracle WebLogic Server on hosts that are running IPv6, you configure `mod_wl_ohs`. You edit the configuration files in the following directory:

DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs1

In the files, specify either the resolvable host name or the IPv6 address in one of the following parameters:

```
WebLogicHost#hostname#|[IPAddress]
WebCluster#[IPAddress_1]:portnum1,#[IPAddress_2]:portnum2,#[IPAddress_3]:portnum3,#
...
```

You must enclose the IPv6 address in brackets.

Any errors are logged in the Oracle HTTP Server logs. To generate more information, set the `mod_weblogic` directives `Debug All` and `WLogFile` path. Oracle HTTP Server logs module-specific messages.

Note: In previous versions, Oracle HTTP Server contained restrictions about using dynamic clusters with IPv6 nodes. For example, the Oracle HTTP Server plug-in for Oracle WebLogic Server had limited IPv6 support in that the DSL (dynamic server list) feature of the plug-in was not supported; only the static configuration of server lists was supported (`DynamicServerList=OFF`). Those restrictions have been lifted.

Part VII

Advanced Administration: Backup and Recovery

Backup and recovery refers to the various strategies and procedures involved in guarding against hardware failures and data loss, and reconstructing data should loss occur. This part describes how to back up and recover Oracle Fusion Middleware.

It contains the following chapters:

- [Chapter 15, "Introducing Backup and Recovery"](#)
- [Chapter 16, "Backing Up Your Environment"](#)
- [Chapter 17, "Recovering Your Environment"](#)

Introducing Backup and Recovery

This chapter provides an introduction to backing up and recovering Oracle Fusion Middleware, including backup and recovery recommendations for Oracle Fusion Middleware components.

It contains the following sections:

- [Understanding Oracle Fusion Middleware Backup and Recovery](#)
- [Oracle Fusion Middleware Directory Structure](#)
- [Overview of the Backup Strategies](#)
- [Overview of Recovery Strategies](#)
- [Backup and Recovery Recommendations for Oracle Fusion Middleware Components](#)
- [Assumptions and Restrictions](#)

15.1 Understanding Oracle Fusion Middleware Backup and Recovery

The installations of an Oracle Fusion Middleware environment are interdependent in that they contain configuration information, applications, and data that are kept in synchronization. For example, when you perform a configuration change, information in configuration files is updated. When you deploy an application, you might deploy it to all Managed Servers in a domain or cluster.

It is, therefore, important to consider your entire Oracle Fusion Middleware environment when performing backup and recovery. You should back up your entire Oracle Fusion Middleware environment at once, then periodically. If a loss occurs, you can restore your environment to a consistent state.

The following topics describe concepts that are important to understanding backup and recovery:

- [Impact of Administration Server Failure](#)
- [Managed Server Independence \(MSI\) Mode](#)
- [Configuration Changes in Managed Servers](#)

See Also: *Understanding Oracle Fusion Middleware* for conceptual information about Oracle WebLogic Server domains, the Administration Server, Managed Servers and clusters, and Node Manager.

15.1.1 Impact of Administration Server Failure

The failure of an Administration Server does not affect the operation of Managed Servers in the domain but it does prevent you from changing the domain's configuration. If an Administration Server fails because of a hardware or software failure on its host computer, other server instances on the same computer may be similarly affected.

If an Administration Server for a domain becomes unavailable while the server instances it manages—clustered or otherwise—are running, those Managed Servers continue to run. Periodically, these Managed Servers attempt to reconnect to the Administration Server. For clustered Managed Server instances, the load balancing and failover capabilities supported by the domain configuration continue to remain available.

When you first start a Managed Server, it must be able to connect to the Administration Server to retrieve a copy of the configuration. Subsequently, you can start a Managed Server even if the Administration Server is not running. In this case, the Managed Server uses a local copy of the domain's configuration files for its starting configuration and then periodically attempts to connect with the Administration Server. When it does connect, it synchronizes its configuration state with that of the Administration Server.

15.1.2 Managed Server Independence (MSI) Mode

A Managed Server maintains a local copy of the domain configuration. When a Managed Server starts, it contacts its Administration Server to retrieve any changes to the domain configuration that were made since the Managed Server was last shut down. If a Managed Server cannot connect to the Administration Server during startup, it can use its locally cached configuration information—this is the configuration that was current at the time of the Managed Server's most recent shutdown. A Managed Server that starts without contacting its Administration Server to check for configuration updates is running in Managed Server Independence (MSI) mode. By default, MSI mode is enabled. However a Managed Server cannot be started even in MSI mode for the first time if the Administration Server is down due to non-availability of the cached configuration.

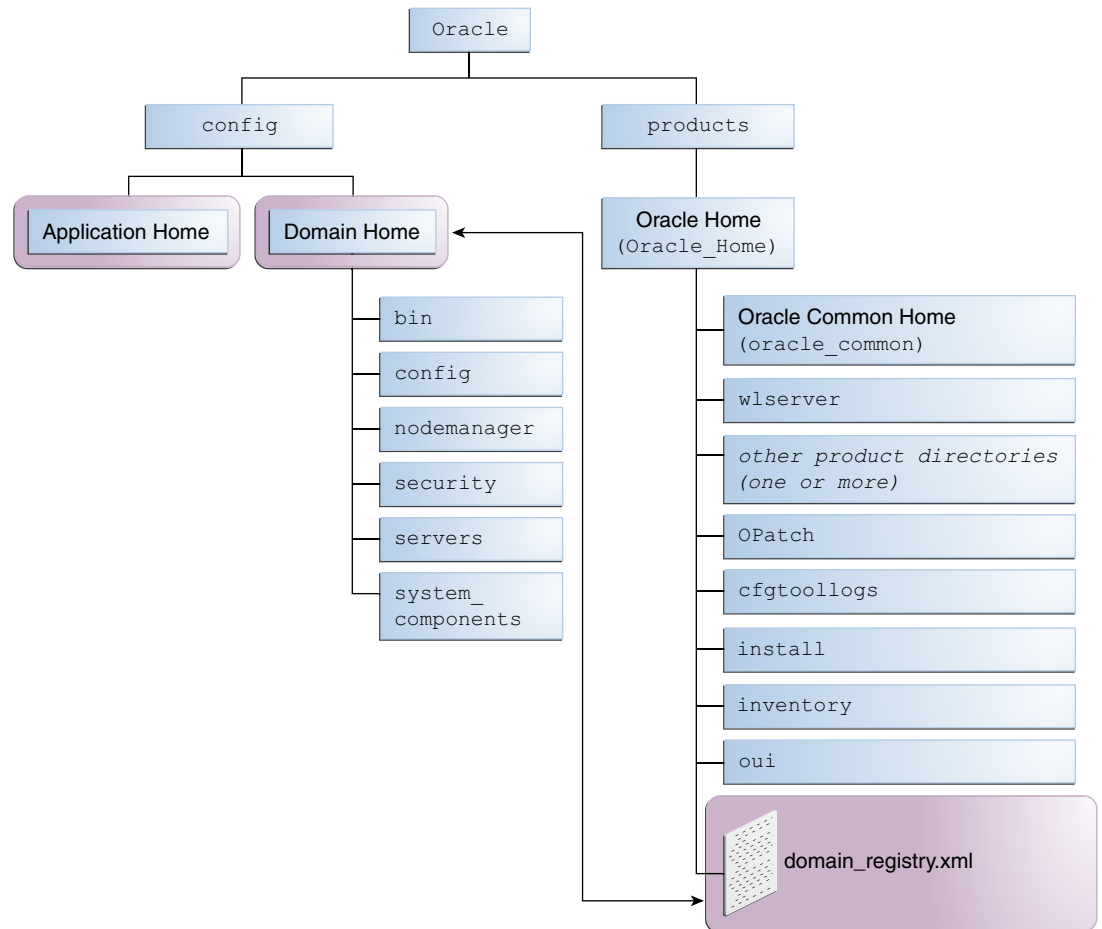
15.1.3 Configuration Changes in Managed Servers

Configuration changes are updated in a Managed Server during the following events:

- On each Managed Server restart, the latest configuration is retrieved from the Administration Server. This happens even when Node Manager is down on the node where the Managed Server is running. If the Administration Server is unavailable during the Managed Server restart and if the MSI (Managed Server Independence) mode is enabled in the Managed Server, it starts by reading its local copy of the configuration and synchronizes with the Administration Server when it is available. By default MSI mode is enabled.
- Upon activating every administrative change such as configuration changes, deployment or redeployment of applications, and topology changes, the Administration Server pushes the latest configuration to the Managed Server. If the Managed Server is not running, the Administration Server pushes the latest version of the configuration to the Managed Server when it does start.

15.2 Oracle Fusion Middleware Directory Structure

The following shows a simplified view of the Oracle Fusion Middleware directory structure when you have installed and configured the Oracle Fusion Middleware Infrastructure:



15.3 Overview of the Backup Strategies

To back up your Oracle Fusion Middleware environment, you can use:

- File copy utilities such as `copy`, `xcopy`, `tar`, or `jar`. Make sure that the utilities:
 - Preserve symbolic links
 - Support long file names
 - Preserve the permissions and ownership of the files

For example:

- On Windows, for online backups, use `copy`; for offline backups, use `copy`, `xcopy`, or `jar`. Do not use Winzip because it does not work with long filenames or extensions.

Note that for some versions of Windows, any file name with more than 256 characters fails. You can use the `xcopy` command with the following switches to work around this issue:

```
xcopy# /s/e##"C:\Temp\*.*"##"C:\copy"
```

See the xcopy help for more information about syntax and restrictions.

- On Linux and UNIX, for online and offline backups, use tar.
- Oracle Recovery Manager (RMAN) to back up database-based metadata repositories and any databases used by Oracle Fusion Middleware. With RMAN, you can perform full backups or incremental backups. See *Oracle Database Backup and Recovery User's Guide* for information about using RMAN to back up a database.

If you want to retain your backups for a longer duration, you may want to back up to tape, for example using Oracle Secure Backup.

You can also configure Oracle WebLogic Server to make backup copies of the configuration files. This facilitates recovery in cases where configuration changes need to be reversed or in the unlikely case that configuration files become corrupted. When the Administration Server starts, it saves a .jar file named config-booted.jar that contains the configuration files. When you make changes to the configuration files, the old files are saved in the configArchive directory under the domain directory, in a .jar file with a sequentially numbered name such as config-1.jar. However, the configuration archive is always local to the Administration Server host. It is a best practice to back up the archives to an external location.

15.3.1 Types of Backups

You can back up your Oracle Fusion Middleware environment offline or online:

- An **offline backup** means that you must shut down the environment before backing up the files. When you perform an offline backup, the Administration Server, all Managed Servers in the domain should be shut down.

Back up the environment offline immediately after installation and after applying any patches or upgrades.
- An **online backup** means that you do not shut down the environment before backing up the files. To avoid an inconsistent backup, do not make any configuration changes until the backup is completed. To ensure that no changes are made in the WebLogic Server domain, lock the WebLogic Server configuration, as described in [Section 2.3.2](#).

You can perform backups on your full Oracle Fusion Middleware environment, or on the run-time artifacts, which are those files that change frequently.

To perform a full backup, you should back up the static files and directories, as well as run-time artifacts, which are described in [Section 15.3.2](#).

15.3.2 Backup Artifacts

Backup artifacts include static files and directories and run-time artifacts.

Static files and directories are those that do not change frequently. These include:

- The Oracle home (*ORACLE_HOME*). An Oracle home consists of product homes, such as the WebLogic Server home and an Oracle Common home. It can also contain the user_projects directories, which contains Oracle WebLogic Server domains, which are not static files.
- OraInventory
- On Linux and UNIX, the oraInst.loc file, which is located in the following directory:

```
(Linux#and#IBM#AIX) #/etc
(Other#UNIX#systems) #/var/opt/oracle
```

- On Linux and UNIX, the `oratab` file, which is located in the following directory:

```
/etc
```

- The `beahomelist` file, which is located at:

```
(UNIX) #user_home/bea/beahomelist
(Windows) #C:\bea\beahomelist
```

- On Windows, the following registry key:

```
HKEY_LOCAL_MACHINE\Software\oracle
```

In addition, for system components, such as Oracle HTTP Server, you must back up the following Windows Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
```

Run-time artifacts are those files that change frequently. Back up these files when you perform a full backup and on a regular basis. Run-time artifacts include:

- Domain directories of the Administration Server and the Managed Servers (by default, a domain directory resides in Oracle home, but it can be configured by the user to point to a different location.)

In most cases, you do not need to back up Managed Server directories separately because the Administration Server contains information about all of the Managed Servers in its domain.

- Application artifacts, such as `.ear` or `.war` files that reside outside of the domain. You do not need to back up application artifacts in a Managed Server directory structure because they can be retrieved from the Administration Server during Managed Server startup.
- Database artifacts, such as the MDS Repository.
- Any database-based metadata repositories used by Oracle Fusion Middleware. You use Oracle Recovery Manager (RMAN) to back up an Oracle database.
- Persistent stores, such as JMS Providers and transaction logs, which reside by default in the `user_projects` directory, but can be configured in a different location.

15.3.3 Recommended Backup Strategy

This section outlines the recommended strategy for performing backups. Using this strategy ensures that you can perform the recovery procedures in this book.

- **Perform a full offline backup:** This involves backing up the entities described in [Section 15.3.2](#). Perform a full offline backup at the following times:
 - Immediately after you install Oracle Fusion Middleware
 - Immediately before upgrading your Oracle Fusion Middleware environment
 - Immediately after an operating system software upgrade
 - Immediately after upgrading or patching Oracle Fusion Middleware
- **Perform an online backup of run-time artifacts:** This involves backing up the run-time artifacts described in [Section 15.3.2](#). Backing up the run-time artifacts enables you to restore your environment to a consistent state as of the time of your

most recent configuration and metadata backup. To avoid an inconsistent backup, do not make any configuration changes until backup completes. Perform an online backup of run-time artifacts at the following times:

- On a regular basis. Oracle recommends that you back up run-time artifacts nightly.
 - Prior to making configuration changes to a component.
 - After making configuration changes to a component.
 - Prior to deploying a custom Java EE application to a Managed Server or cluster.
 - After a major change to the deployment architecture, such as creating servers or clusters.
- **Perform a full or incremental backup of your databases:** Use RMAN to backup your databases. See the *Oracle Database Backup and Recovery User's Guide* for information about using RMAN and for suggested methods of backing up the databases.

15.4 Overview of Recovery Strategies

Recovery strategies enable you to recover from critical failures that involve actual data loss. Depending on the type of loss, they can involve recovering any combination of the following types of files:

- Oracle software files
- Configuration files
- Oracle system files
- Windows Registry keys
- Application artifacts

You can recover your Oracle Fusion Middleware environment while Oracle Fusion Middleware is offline.

To recover your Oracle Fusion Middleware environment, you can use:

- File copy utilities such as copy, xcopy, or tar

When you restore the files, use your preferred tool to extract the compressed files:

- On Windows, for online recovery, use copy; for offline recovery, use copy, xcopy, or jar.

Note that for some versions of Windows, any file name with more than 256 characters fails. You can use the xcopy command with the following switches to work around this issue:

```
xcopy# /s/e##"C:\Temp\*. * ##"C:\copy"
```

See the xcopy help for more information about syntax and restrictions.

Do not use Winzip because it does not work with long filenames or extensions.

- On Linux and UNIX, use tar.

Ensure that the tool you are using preserves the permissions and timestamps of the files.

- Oracle Recovery Manager (RMAN) to recover database-based metadata repositories

15.4.1 Types of Recovery

You can recover your Oracle Fusion Middleware environment in part or in full. You can recover the following:

- The Oracle home
- WebLogic Server domains
- The WebLogic Server Administration Server
- WebLogic Server Managed Servers
- A component, such as Oracle HTTP Server
- WebLogic Server cluster
- Deployed applications

15.4.2 Recommended Recovery Strategies

Note the following key points about recovery:

- Your Oracle Fusion Middleware environment must be offline while you are performing recovery.
- Rename important existing files and directories before you begin restoring the files from backup so that you do not unintentionally override necessary files.
- Although, in some cases, it may appear that only one or two files are lost or corrupted, you should restore the directory structure for the entire element, such as a domain, rather than just restoring one or two files. In this way, you are more likely to guarantee a successful recovery.
- Recover the database to the most current state, using point-in-time recovery (if the database is configured in Archive Log Mode). This is typically a time right before the database failure occurred.

15.5 Backup and Recovery Recommendations for Oracle Fusion Middleware Components

The following sections describe backup and recovery recommendations for specific Oracle Fusion Middleware components:

- [Backup and Recovery Recommendations for Oracle WebLogic Server](#)
- [Backup and Recovery Recommendations for Oracle JRF Installations](#)
- [Backup and Recovery Recommendations for Web Tier Installations](#)
- [Backup and Recovery Recommendations for Oracle Data Integrator](#)

For the steps you take to back up your environment, see [Section 16.3](#). For the steps you take to recover your environment, see [Chapter 17](#).

15.5.1 Backup and Recovery Recommendations for Oracle WebLogic Server

The following sections describe backup and recovery recommendations for Oracle WebLogic Server:

- [Backup and Recovery Recommendations for Oracle WebLogic Server](#)
- [Backup and Recovery Recommendations for Oracle WebLogic Server JMS](#)

15.5.1.1 Backup and Recovery Recommendations for Oracle WebLogic Server

This section describes the Oracle WebLogic Server data that must be backed up and restored.

Database Repository Dependencies

Oracle WebLogic Server does not, by default, depend on any database repository. However, applications deployed on Oracle WebLogic Server may use databases as data sources. To back up a database, see the *Oracle Database Backup and Recovery User's Guide*.

Backup Recommendations

Back up the Oracle home and the domain.

Recovery Recommendations

Depending on what has failed, you may need to recover the following:

- The domain: See [Section 17.2.2](#).
- The Administration Server configuration: See [Section 17.2.4](#).
- A Managed Server: See [Section 17.2.5](#).
- A cluster: See [Section 17.2.7](#).
- Applications: See [Section 17.2.8](#).

If you use Whole Server Migration, the leasing information is stored in a table in a database. If you recover Oracle WebLogic Server, you should discard the information in the leasing table. (For more information about Whole Server Migration, see "Whole Server Migration" in *Administering Clusters for Oracle WebLogic Server*.)

After a loss of host, you may need to recover the following:

- The WebLogic domain. See [Section 17.3.1](#).
- The Administration Server host: See [Section 17.3.3](#).
- The Managed Server host: See [Section 17.3.4](#).

15.5.1.2 Backup and Recovery Recommendations for Oracle WebLogic Server JMS

This section describes the Oracle WebLogic Server JMS data that must be backed up and restored.

Database Repository Dependencies

Only if JMS is database-based

Backup Recommendations

Back up the domain.

If you are using a database-based JMS, back up the database using RMAN.

If you are using file-based JMS, use storage snapshot techniques for taking consistent online backups. Alternatively, you can use a file system copy to perform an offline backup.

Recovery Recommendations

Recover the domain.

If the JMS persistent store is file-based, recover it from backup. If the JMS persistent store is database-based, recover the database to the most recent point in time, if needed. Note the following:

- Always try to keep JMS data as current as possible. This can be achieved by using the point-in-time recovery capabilities of Oracle Database, recovering to the most recent time (in the case of database-based persistence) or using a highly available RAID-backed storage device (for example, SAN/NAS).
- If you are using a file-based JMS, you can use storage snapshots to recover.
- If, for whatever reason, you need to restore JMS data to a previous point in time, there are potential implications. Restoring the system state to a previous point in time not only can cause duplicate messages, but can also cause lost messages. The lost messages are messages that were enqueued before or after the system restore point time, but never processed.

Use the following procedure *before recovery* to drain messages in the JMS queue after persistent-store recovery to avoid processing duplicate messages:

Note: Do not drain and discard messages without first being certain that the messages contain no data that must be preserved. The recovered messages may include unprocessed messages with important application data, in addition to duplicate messages that have already been processed.

1. Log into the Oracle WebLogic Server Administration Console.
2. Before recovery, configure JMS server to pause Production, Insertion, and consumption operations at boot time to ensure that no new messages are produced or inserted into the destination or consumed from the destination before you drain stale messages. To do this:
 - a. Expand **Services**, then **Messaging**, and then click **JMS Servers**.
 - b. On the Summary of JMS Servers page, click the JMS server you want to configure for message pausing.
 - c. On the Configuration: General page, click **Advanced** to define the message pausing options. Select **Insertion Paused At Startup**, **Production Paused At Startup**, and **Consumption Paused At Startup**.
 - d. Click **Save**.

Use the following procedure *after recovery*:

1. After recovering the persistent store, start the Managed Servers.
2. Drain the stale messages from JMS destinations, by taking the following steps:
 - a. Expand **Services**, then **Messaging**, and then **JMS Modules**.
 - b. Select a JMS module, then select a target.
 - c. Select **Monitoring**, then **Show Messages**.
3. Click **Delete All**.
4. Resume operations, by taking the following steps:

- a. Expand **Services**, then **Messaging**, and then **JMS Servers**.
- b. On the Summary of JMS Servers page, click the JMS server you want to configure for message pausing.
- c. On the Configuration: General page, click **Advanced**. Deselect **Insertion Paused At Startup**, **Production Paused At Startup**, and **Consumption Paused At Startup**.
- d. Click **Save**.

If the store is not dedicated to JMS use, use the Oracle WebLogic Server JMS message management administrative tool. This tool can perform import, export, move, and delete operations from the Administration Console, MBeans, and WLST.

For applications that use publish and subscribe in addition to queuing, you should manipulate topic subscriptions in addition to queues.

For the steps to recover the domain, see [Section 17.2.2](#) and [Section 17.3.1](#).

15.5.2 Backup and Recovery Recommendations for Oracle JRF Installations

The following topics describe backup and recovery recommendations for components that are installed with more than one type of installation:

- [Backup and Recovery Recommendations for Oracle Web Services Manager](#)
- [Backup and Recovery Recommendations for Oracle Platform Security Services](#)

15.5.2.1 Backup and Recovery Recommendations for Oracle Web Services Manager

This section describes the Oracle Web Services Manager data that must be backed up and restored.

Database Repository Dependencies

If a database-based MDS Repository is used, Oracle Web Services Manager uses a partition in the MDS schema.

Backup Recommendations

Back up the Oracle Web Services Manager domain.

If Oracle Web Services Manager uses a file-based MDS Repository, back it up using a file copy mechanism. If it uses a database-based MDS Repository, back up the database using RMAN.

Recovery Recommendations

Restore the Oracle Web Services Manager Managed Server.

If Oracle Web Services Manager uses a file-based MDS Repository, restore it from the backup. If it uses a database-based MDS Repository, recover the database to the most recent point in time, if needed.

For the steps to recover components, see [Section 17.2.6](#). For the steps specific to recovering from loss of host, see [Section 17.3.5](#).

15.5.2.2 Backup and Recovery Recommendations for Oracle Platform Security Services

This section describes the Oracle Platform Security Services data that must be backed up and restored.

Database Repository Dependencies

If a database-based Oracle Platform Security Repository is used, Oracle Platform Security uses a partition in the OPSS schema.

If an Oracle Internet Directory based Oracle Platform Security repository is used, Oracle Platform Security uses Oracle Internet Directory.

Backup Recommendations

Back up the Administration Server domain, including the following files:

```
DOMAIN_HOME/config/fmwconfig/jps-config.xml
DOMAIN_HOME/config/fmwconfig/jps-config-jse.xml
DOMAIN_HOME/config/fmwconfig/cwallet.sso
DOMAIN_HOME/config/fmwconfig/bootstrap/cwallet.sso
DOMAIN_HOME/config/fmwconfig/keystores.xml
DOMAIN_HOME/config/config.xml
DOMAIN_HOME/config/fmwconfig/ids_config.xml
DOMAIN_HOME/config/fmwconfig/system-jazn-data.xml#(if#present)
DOMAIN_HOME/config/fmwconfig/jps_mbeans.xml
```

Back up Oracle Internet Directory if Oracle Platform Security uses an Oracle Internet Directory based repository.

Backup the database containing the OPSS schema if Oracle Platform Security uses a database-based repository.

Recovery Recommendations

Restore the jps-config.xml file.

If Oracle Platform Security uses a database-based repository, restore the database to the most recent point in time.

For the steps to recover components, see [Section 17.2.6](#). For the steps specific to recovering from loss of host, see [Section 17.3.5](#).

15.5.3 Backup and Recovery Recommendations for Web Tier Installations

The following sections describe backup and recovery recommendations for Web Tier installations:

- [Backup and Recovery Recommendations for Oracle HTTP Server](#)

15.5.3.1 Backup and Recovery Recommendations for Oracle HTTP Server

This section describes the Oracle HTTP Server data that must be backed up and restored.

Database Repository Dependencies

None

Backup Recommendations

For Oracle HTTP Server in a standalone domain, back up the domain. For Oracle HTTP Server in a WebLogic Server domain, back up the domain.

Recovery Recommendations

For Oracle HTTP Server in a standalone domain, restore the domain that contains Oracle HTTP Server, as described in [Section 17.2.3](#). For loss of host, restore the domain, as described in [Section 17.3.2](#) and make configuration changes as described in [Section 17.3.5.5.1](#) or [Section 17.3.5.5.2](#).

For Oracle HTTP Server in a WebLogic Server domain, restore the domain, as described in [Section 17.2.2](#). For loss of host, restore the domain as described in [Section 17.3.1](#) and make configuration changes as described in [Section 17.3.5.5.2](#).

15.5.4 Backup and Recovery Recommendations for Oracle Data Integrator

This section describes the Oracle Data Integrator data that must be backed up and restored.

Database Repository Dependencies

ODI_REPO schema

Backup Recommendations

Back up the Oracle home, domain if Oracle Data Integrator is installed in a domain, and the *ODI_Oracle_Home/oracledi/agent* folder for each machine where a standalone agent is installed.

Back up the database containing Oracle Data Integrator schema.

Recovery Recommendations

Recover the database to the most recent point in time, if needed.

Depending on the extent of the failure and the type of Oracle Data Integrator installation, restore the Managed Server or the Oracle home, or both.

If your environment contains the Oracle Data Integrator Standalone Agent, restore the Oracle home, as described in [Section 17.2.1](#).

If your environment contains Oracle Data Integrator for Developers, restore the Oracle home, as described in [Section 17.2.1](#).

If your environment contains Oracle Data Integrator deployed in a Managed Server, restore the Managed Server, as described in [Section 17.2.5](#).

For the steps to recover Oracle Data Integrator from loss of host, see [Section 17.3.5.6](#).

15.6 Assumptions and Restrictions

The following assumptions and restrictions apply to the backup and recovery procedures in this book. Also see the restrictions listed in [Section 16.2](#).

- Only the user who installs the product or a user who has access privileges to the directories where Oracle Fusion Middleware has been installed should be able to execute backup and recovery operations.
- If a single Managed Server and Administration Server run on different hosts and the Managed Server is not in a cluster, you must use the pack and unpack commands on the Managed Server to retrieve the correct configuration.

See Also: If you are using Cold Failover Cluster or Disaster Recovery, refer to the *Oracle Fusion Middleware Disaster Recovery Guide* for additional information.

Backing Up Your Environment

This chapter describes recommended backup strategies for Oracle Fusion Middleware and the procedures for backing up Oracle Fusion Middleware.

This chapter includes the following sections:

- [Overview of Backing Up Your Environment](#)
- [Limitations and Restrictions for Backing Up Data](#)
- [Performing a Backup](#)
- [Creating a Record of Your Oracle Fusion Middleware Configuration](#)

16.1 Overview of Backing Up Your Environment

As described in [Section 15.3.3](#), you should use the following recommended strategy for backing up your Oracle Fusion Middleware environment:

- If you are performing an online backup, do not make any configuration changes until the backup is completed. To ensure that no changes are made in the WebLogic Server domain, lock the WebLogic Server configuration, as described in [Section 2.3.2](#).
- Perform a full offline backup immediately after you install Oracle Fusion Middleware. See [Section 16.3.1](#) for information on performing a full backup.
- Perform backups of run-time artifacts after every administrative change and on a regular basis. Oracle recommends that you back up run-time artifacts nightly. See [Section 16.3.2](#) for information on performing a backup of run-time artifacts.
- Perform a new full backup after a major change, such as any upgrade or patch, or if any of the following files are modified:

```
DOMAIN_HOME/nodemanager/nodemanager.properties
ORACLE_HOME/wlserver/common/bin/wlsifconfig.sh
ORACLE_HOME/wlserver/common/bin/commEnv.sh
```

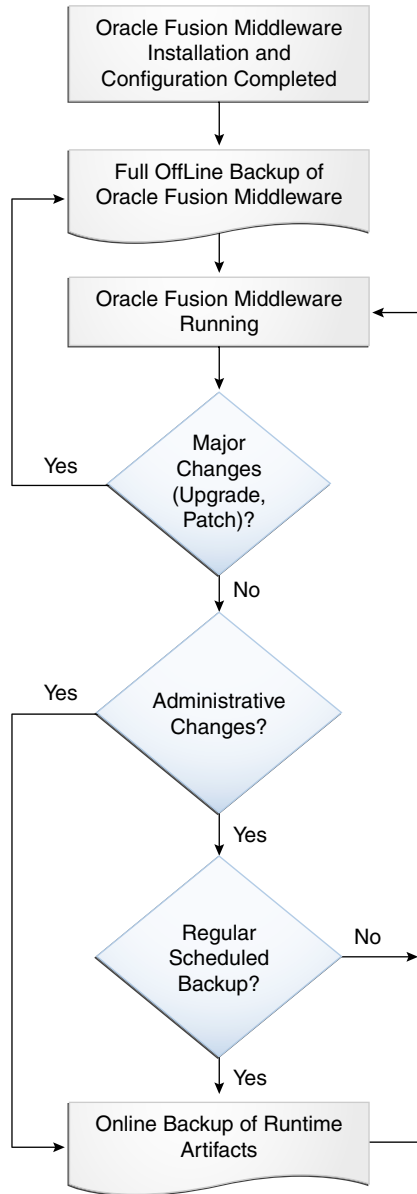
See [Section 16.3.1](#) for information on performing a full backup.

- Create a record of your Oracle Fusion Middleware environment. See [Section 16.4](#).
- When you create the backup, name the archive file with a unique name. Consider appending the date and time to the name. For example, if you create a backup of the Oracle home on April 20, 2013, name the backup:

```
oracle_home_backup_042013.tar
```

The flowchart in [Figure 16–1](#) provides an overview of how to decide which type of backup is appropriate for a given circumstance.

Figure 16–1 Decision Flow Chart for Type of Backup



16.2 Limitations and Restrictions for Backing Up Data

Note the following points:

- LDAP backups: If you use the built-in LDAP, do not update the configuration of a security provider while a backup of LDAP data is in progress. If a change is made (for example, if an administrator adds a user), while you are backing up the ldap directory tree, the backups in the ldapfiles subdirectory could become inconsistent. Refer to *WebLogic Server Managing Server Startup and Shutdown* for detailed LDAP backup procedures.

- Java Transaction API (JTA): Oracle does not recommend that you back up and restore JTA transaction logs.
- Audit Framework: If you have configured Oracle Fusion Middleware Audit Framework to write data to a database, you should not back up the local files in the bus stop. (Auditable events from each component are stored in a repository known as a bus stop; each Oracle WebLogic Server has its own bus stop. Data can be persisted in this file, or uploaded to a central repository at which point the records are available for viewing and reporting.)

If you back up the local files, duplicate records are uploaded to the database. That is, they are uploaded to the database when the bus stop is created and then are uploaded again when you restore the files.

The default locations for bus stop local files are:

- For Java components:

```
DOMAIN_HOME/servers/server_name/logs/auditlogs/component_type
```

- For system components, such as Oracle HTTP Server:

```
DOMAIN_HOME/auditlogs/component_type/component_name
```

For more information about Oracle Fusion Middleware Audit Framework and the bus stop, see "Configuring and Managing Auditing" in *Securing Applications with Oracle Platform Security Services*.

16.3 Performing a Backup

You can perform a full offline backup or an online or offline backup of run-time artifacts, as described in the following topics:

- [Performing a Full Offline Backup](#)
- [Performing an Online Backup of Run-Time Artifacts](#)
- [Backing Up Windows Registry Entries](#)

16.3.1 Performing a Full Offline Backup

To perform a full offline backup, you copy the directories that contain Oracle Fusion Middleware files.

Archive and compress the source Oracle home, using your preferred tool for archiving, as described in [Section 15.3](#).

Take the following steps:

1. Shut down all processes in the Oracle home. For example, shut down the Managed Servers, the Administration Server, and any system components.
2. Back up the Oracle home (ORACLE_HOME) on all hosts. For example:

```
(UNIX) #tar#-cf#oracle_home_backup_042013.tar#ORACLE_HOME/*
(Windows) #jar#cMf#oracle_home_backup_042013.jar#ORACLE_HOME\*
```

3. Back up the Administration Server domain separately. This backs up Java components and any system components in the domain.

For example:

```
(UNIX) #tar#-cf#domain_home_backup_042013.tar#DOMAIN_HOME/*
(Windows) #jar#cMf#domain_home_backup_042013.jar#DOMAIN_HOME\*
```

##

In most cases, you do not need to back up the Managed Server directories separately, because the Administration Server domain contains information about the Managed Servers in its domain. If you have customized your environment for the Managed Server, back up the Managed Server directories. See [Section 15.5](#) for information about what you need to back up.

4. If a Managed Server is not located within the domain, back up the Managed Server directory. For example:

```
(UNIX) #tar#-cf#mg1_home_backup_042013.tar#server_name/*
(Windows) #jar#cmf#mg1_home_backup_042013.jar#server_name\*
```

5. Back up the application home directory. For example:

```
(UNIX) #tar#-cf#app_home_backup_042013.tar#Applications_Home/domain_name/*
(Windows) #jar#cmf#app_home_backup_042013.jar#Applications_Home\domain_name\*
```

6. Back up the OraInventory directory. For example:

```
tar#-cf#Inven_home_backup_042013.tar#/scratch/oracle/OraInventory
```

7. On Linux and UNIX, back up the oraInst.loc file, which is located in the following directory:

```
(Linux#and#IBM#AIX) #/etc
(Other#UNIX#systems) #/var/opt/oracle
```

8. On Linux and UNIX, backup the oratab file, which is located in the following directory:

```
/etc
```

9. Back up the database repositories using the Oracle Recovery Manager (RMAN). For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*.
10. On Windows, export the Windows Registry entries, as described in [Section 16.3.3](#).
11. Unlock the WebLogic Server configuration by clicking Release Configuration on the WebLogic Server Administration Console,
12. Create a record of your Oracle Fusion Middleware environment. See [Section 16.4](#).

16.3.2 Performing an Online Backup of Run-Time Artifacts

You should perform a backup of run-time artifacts (which are listed in [Section 15.3.2](#)) on a regular basis and at the times described in [Section 15.3.3](#).

To back up run-time artifacts:

1. To avoid an inconsistent backup, do not make any configuration changes until the backup is completed. To ensure that no changes are made in the WebLogic Server domain, lock the WebLogic Server configuration, as described in [Section 2.3.2](#).
2. Back up the Administration Server domain directories. For example:

```
UNIX) #tar#-cf#domain_home_backup_042013.tar#
/scratch/oracle/config/domains/domain_name/*
(Windows) #jar#cmf#domain_home_backup_042013.jar#
C:\oracle\config\domains\domain_name\*#
```

3. Back up the application home directory. For example:

```
(UNIX) #tar#-cf#app_home_backup_042013.tar#/scratch/oracle/applications/domain_
```

```

name/*
(Windows)#jar#cmf#app_home_backup_042013.jar#C:\oracle\applications\domain_
name\*
##

```

4. Back up the database repositories using the Oracle Recovery Manager (RMAN). For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*.
5. Unlock the Oracle WebLogic Server configuration by clicking **Release Configuration** on the WebLogic Server Administration Console,
6. Create a record of your Oracle Fusion Middleware environment. See [Section 16.4](#).

16.3.3 Backing Up Windows Registry Entries

On Windows, you must back up Windows Registry keys related to Oracle Fusion Middleware. Which keys you back up depends on what components you have installed.

To export a key, use the following command:

```
regedit#/E##FileName#Key
```

Export the following entries:

- For any component, export the following registry key:

```
HKEY_LOCAL_MACHINE\Software\Oracle
```

- For system components, such as Oracle HTTP Server, export each node that begins **Oracle** within the following registry keys:

```

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services

```

For example:

```
regedit#/E#C:\oracleSMP.reg#HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services
```

Use a unique file name for the each key.

You can also use the Registry Editor to export the key. See the Registry Editor Help for more information.

16.4 Creating a Record of Your Oracle Fusion Middleware Configuration

In the event that you need to restore and recover your Oracle Fusion Middleware environment, it is important to have all the necessary information at your disposal. This is especially true in the event of a hardware loss that requires you to reconstruct all or part of your Oracle Fusion Middleware environment on a new disk or host.

You should maintain an up-to-date record of your Oracle Fusion Middleware environment that includes the information listed in this section. You should keep this information both in hardcopy and electronic form. The electronic form should be stored on a host or e-mail system that is completely separate from your Oracle Fusion Middleware environment.

Your Oracle Fusion Middleware hardware and software configuration record should include:

- The following information for each host in your environment:
 - Host name

- Virtual host name (if any)
- Domain name
- IP address
- Hardware platform
- Operating system release level and patch information
- The version of the JDK and its path used in the installation and configuration of Oracle Fusion Middleware
- The following information for each Oracle Fusion Middleware installation in your environment:
 - Installation type (for example, Oracle HTTP Server)
 - Host on which the installation resides
 - User name, userid number, group name, groupid number, environment profile, and type of shell for the operating system user that owns the Oracle home (`/etc/passwd` and `/etc/group` entries)
 - Directory structure, mount points, and full path for the Oracle home, Oracle Common home, product homes, Oracle WebLogic Server domain home (if it does not reside in the `user_projects` directory in the Oracle home)
 - Amount of disk space used by the installation
 - Port numbers used by the installation
 - The version of the JDK and its path used in the installation and configuration of Oracle Fusion Middleware
- The following information for the database containing the metadata for components:
 - Host name
 - Database version and patch level
 - Base language
 - Character set
 - Global database name
 - SID
 - Listen port

Recovering Your Environment

This chapter describes recommended recovery strategies and procedures for recovering Oracle Fusion Middleware from different types of failures and outages, such as media failures or loss of host.

This chapter includes the following sections:

- [Overview of Recovering Your Environment](#)
- [Recovering After Data Loss, Corruption, Media Failure, or Application Malfunction](#)
- [Recovering After Loss of Host](#)

17.1 Overview of Recovering Your Environment

This section provides an overview of recovery strategies for outages that involve actual data loss or corruption, host failure, or media failure where the host or disk cannot be restarted and they are permanently lost. This type of failure requires some type of data restoration before the Oracle Fusion Middleware environment can be restarted and continue with normal processing.

Note: The procedures in this chapter assume that no administrative changes were made since the last backup. If administrative changes were made since the last backup, they must be reapplied after recovery is complete.

When you restore the files, use your preferred tool to extract the compressed files, as described in [Section 15.4](#).

Ensure that the tool you are using preserves the permissions and timestamps of the files.

Rename existing files and directories before you begin restoring the files from backup so that you do not unintentionally override necessary files.

When you recover your environment, it is important to recover the entities in the correct order:

1. The database, if it needs to be recovered. See [Section 17.2.9](#) and [Section 17.3.7](#).
2. The Oracle Home, if it needs to be recovered. See [Section 17.2.1](#).
3. The entire domain, if it needs to be recovered. See [Section 17.2.2](#) and [Section 17.3.1](#) for recovering a WebLogic Server managed domain. See [Section 17.2.3](#) for recovering a standalone domain.

4. The Administration Server, if you do not need to recover the domain. See [Section 17.2.4](#) and [Section 17.3.3](#),
5. The Managed Servers, if they are not in the Administration Server domain directory and they need to be recovered. See [Section 17.2.5](#) and [Section 17.3.4](#).
Java components are recovered when you recover the Managed Server. System components are recovered when you recover the domain.
However, when you have a loss of host, you may need to take additional steps to recover components. See [Section 17.3.5](#)
6. Applications, if they need to be recovered. See [Section 17.2.8](#).

17.2 Recovering After Data Loss, Corruption, Media Failure, or Application Malfunction

This section describes recovery strategies for outages that involve actual data loss or corruption, or media failure where the disk cannot be restored. It also describes recovery strategies for applications that are no longer functioning properly. This type of failure requires some type of data restoration before the Oracle Fusion Middleware environment can be restarted and continue with normal processing. It contains the following topics:

- [Recovering the Oracle Home](#)
- [Recovering an Oracle WebLogic Server Domain](#)
- [Recovering a Standalone Domain](#)
- [Recovering the Administration Server Configuration](#)
- [Recovering a Managed Server](#)
- [Recovering Components](#)
- [Recovering a Cluster](#)
- [Recovering Applications](#)
- [Recovering a Database](#)

Note: You can only restore an entity to the same path as the original entity. That path can be on the same host or a different host.

17.2.1 Recovering the Oracle Home

You can recover the Oracle home that was corrupted or from which files were deleted.

To recover the Oracle home:

1. Stop all relevant processes. That is, stop all processes that are related to the domain, such as the Administration Server, Node Manager, and Managed Servers. For example, to stop the Administration Server on Linux:

```
DOMAIN_HOME/bin/stopWebLogic.sh#username#password#[admin_url]
```

2. Change to the directory that you want to be the parent directory of the Oracle home directory. Use the same directory structure as in the original environment.
3. Recover the Oracle home directory from backup. For example:

```
(UNIX)#tar#-xf#oracle_home_backup_042013.tar
```



```
(Windows) #jar#xf#oracle_home_backup_042013.jar
```

4. Start all relevant processes. That is, start all processes that run in the Oracle home, such as the Administration Server and Managed Servers. For example, start the Administration Server:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

17.2.2 Recovering an Oracle WebLogic Server Domain

You can recover an Oracle WebLogic Server domain that was corrupted or deleted from the file system.

Caution: Performing a domain-level recovery can impact other aspects of a running system and all of the configuration changes performed after the backup was taken will be lost.

To recover an Oracle WebLogic Server domain that was corrupted or deleted from the file system:

1. Stop all relevant processes. That is, stop all processes that are related to the domain, such as the Administration Server, Managed Servers, and any system components. For example, stop the Administration Server:

```
DOMAIN_HOME/bin/stopWebLogic.sh#username#password#[admin_url]
```

For information on stopping system components such as Oracle HTTP Server, see [Section 4.3.2](#).

2. Change to the directory that you want to be the parent directory of the domain home directory. Use the same directory structure as in the original environment.
3. Recover the domain directory from backup:

```
(UNIX) #tar#-xf#domain_backup_042013.tar#
(Windows) #jar#xf#domain_backup_042013.jar#
```

4. Start all relevant processes. That is, start all processes that are related to the domain, such as the Administration Server, Managed Servers, and any system components. For example, start the Administration Server:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

For information on starting system components such as Oracle HTTP Server, see [Section 4.3.2](#).

5. If you cannot start the Administration Server, recover it, as described in [Section 17.2.4](#).
6. If you cannot start a Managed Server, recover it, as described in [Section 17.2.5](#).

17.2.3 Recovering a Standalone Domain

You can recover a standalone domain that contains system components such as Oracle HTTP Server, that was corrupted or deleted from the file system.

To recover a standalone domain:

1. Stop Node Manager as described in [Section 4.2.2](#).
2. Recover the domain home:

```
(UNIX) #tar#xf#domain_backup_05_21_2013.tar  
(Windows) #jar#xf#domain_backup_05_21_2013.jar
```

3. Start Node Manager:

```
(UNIX) #DOMAIN_HOME/bin/startNodeManager.sh  
(Windows) #DOMAIN_HOME\bin\startNodeManager.cmd
```

4. Start any system components, such as Oracle HTTP Server, that are in the domain:

```
(UNIX) #Domain_Home/bin/startComponent.sh#ohs1  
(Windows) #Domain_Home\bin\startComponent.cmd#ohs1
```

17.2.4 Recovering the Administration Server Configuration

If the Administration Server configuration has been lost because of file deletion or file system corruption, the Administration Server console continues to function if it was already started when the problem occurred. To prevent the Administration Server from prompting for a user name and password, see [Section 4.2.4](#).

Caution: Performing a domain-level recovery can impact other aspects of a running system and all of the configuration changes performed after the backup was taken will be lost.

To recover the Administration Server configuration:

1. Stop all processes, including the Administration Server, Managed Servers, and Node Manager, if they are started. For example, to stop the Administration Server:

```
DOMAIN_HOME/bin/stopWebLogic.sh#username#password#[admin_url]
```

2. Recover the Administration Server configuration by recovering the domain home backup to a temporary location. Then, restore the config directory to the following location:

```
DOMAIN_HOME/config
```

3. Start the Administration Server. For example:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

4. Verify that the Administration Server starts properly and is accessible.

On the next configuration change, the configuration from the Administration Server is pushed to the Managed Servers. On each Managed Server restart, the configuration is retrieved from the Administration Server.

17.2.5 Recovering a Managed Server

You can recover a Managed Server's files, including its configuration files if they are deleted or corrupted.

The following topics describe how to recover a Managed Server's files:

- [Recovering a Managed Server When It Cannot Be Started](#)
- [Recovering a Managed Server When It Does Not Function Correctly](#)

17.2.5.1 Recovering a Managed Server When It Cannot Be Started

In this scenario, the Managed Server is not on the same host as the Administration Server, and it does not operate properly or cannot be started because the configuration has been deleted or corrupted or the configuration was mistakenly changed and you cannot ascertain what was changed.

To recover a Managed Server when it cannot be started:

1. If the Administration Server is not reachable, recover the Administration Server, as described in [Section 17.2.4](#).
2. If the Managed Server fails to start, recover the Oracle home from the backup, if required. For example:

```
tar#-xf#oracle_home_backup_042013.tar#
```

3. Stop Node Manager as described in [Section 4.2.2](#).
4. If the file system is lost, take the following steps:
 - a. Create a domain template jar file for the Administration Server, using the pack utility on the Administration Server host. For example:

```
pack.sh#-domain=/scratch/oracle/config/domains/domain_name
###-template=/scratch/temp.jar#-template_name=test_install#
###-template_author=myname#-log=/scratch/logs/my.log#-managed=true
```

Specifying the `-managed=true` option packs up only the Managed Servers. If you want to pack the entire domain, omit this option.

- b. Unpack the domain template jar file, using the unpack utility on the Managed Server host. In the following example, `temp.jar` is the archive created by the pack command:

```
unpack.sh#-template=/scratch/temp.jar
###-domain=/scratch/oracle/config/domains/domain_name
###-log=/scratch/logs/new.log#-log_priority=info
```

Note:

- The following directory must exist. If it does not the unpack command fails.


```
ORACLE_HOME/config/domains/
```
- The unpack command provides an `-overwrite_domain` option, which allows unpacking a Managed Server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. Use the `-overwrite_domain` option, if required for your deployment.
- By default, applications are stored in the following directory unless you pass another location using the `-app_dir` argument:

```
ORACLE_HOME/user_projects/applications/Domain_Name
```

- c. Ensure that the application artifacts are accessible from the Managed Server host. That is, if the application artifacts are not on the same server as the Managed Server, they must be in a location accessible by the Managed Server.

Note:

- For stage mode applications, the Administration server copies the application bits to the staged directories on the Managed Server hosts.
- For nostage and external_stage mode applications, ensure that application files are available in the stage directories of the Managed Server.

See *Deploying Applications to Oracle WebLogic Server* for information about stage, nostage and external_stage modes.

5. Start the Managed Server. For example:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

The Managed Server connects to the Administration Server and updates its configuration changes.

17.2.5.2 Recovering a Managed Server When It Does Not Function Correctly

In this scenario, the Managed Server is running, but the file system for the Managed Server has been lost or corrupted.

To recover the Managed Server:

1. Stop the Managed Server. For example:

```
DOMAIN_HOME/bin/stopManagedWeblogic.sh#managed_server_name#admin_url
#####username#password
```

2. Recover the Oracle home from the backup, if required:

```
tar#-xf#oracle_home_backup_042013.tar#
```

3. Stop Node Manager as described in [Section 4.2.2](#).

4. Create a domain template jar file for the Administration Server, using the pack utility on the Administration Server host. For example:

```
pack.sh#-domain=/scratch/oracle/config/domains/WLS_domain
##-template=/scratch/temp.jar#-template_name=test_install#
##-template_author=myname#-log=/scratch/logs/my.log#-managed=true
```

Specifying the `-managed=true` option packs up only the Managed Servers. If you want to pack the entire domain, omit this option.

5. Unpack the domain template jar file, using the unpack utility on the Managed Server host. In the following example, `temp.jar` is the archive created by the pack command:

```
unpack.sh#-template=/scratch/temp.jar
###-domain=/scratch/oracle/config/domains/WLS_domain
###-log=/scratch/logs/new.log#-log_priority=info
```

Note:

- The following directory must exist. If it does not the unpack command fails.

ORACLE_HOME/config/domains/

- The unpack command provides an `-overwrite_domain` option, which allows unpacking a Managed Server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. Use the `-overwrite_domain` option, if required for your deployment.
- By default, applications are stored in the following directory unless you pass another location using the `-app_dir` argument:

ORACLE_HOME/user_projects/applications/Domain_Name

6. Ensure that the application artifacts are accessible from the Managed Server host. That is, if the application artifacts are not on the same server as the Managed Server, they must be in a location accessible by the Managed Server.

Note:

- For stage mode applications, the Administration server copies the application bits to the staged directories on the Managed Server hosts.
- For nostage and external_stage mode applications, ensure that application files are available in the stage directories of the Managed Server.

See *Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

7. Restart the Managed Server. For example:

DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url

17.2.6 Recovering Components

For components such as Oracle WebLogic Server and Oracle HTTP Server, you recover the components using the procedures in the following topics:

- [Recovering a Component That Is Not Functioning Properly](#)
- [Recovering a Component After Cluster Configuration Change](#)

17.2.6.1 Recovering a Component That Is Not Functioning Properly

You can recover a component if the component's files have been deleted or corrupted or if the component cannot be started or is not functioning properly because the component's configuration was changed and committed. You may not be able to ascertain what change is causing the problem and you want to revert to an earlier version.

- For Java components, you recover the Managed Server, as described in [Section 17.2.5](#).

- For system components, such as Oracle HTTP Server, in a standalone domain, you recover the domain, as described in [Section 17.2.3](#).
- For system components, such as Oracle HTTP Server, in an Oracle WebLogic Server domain, you recover the domain as described in [Section 17.2.2](#).

17.2.6.2 Recovering a Component After Cluster Configuration Change

You can recover components in a cluster when the components cannot be started or are not functioning properly because the configuration was changed and committed at the cluster level. You may not be able to ascertain what change is causing the problem and you want to revert to an earlier version.

Caution: Performing a domain-level recovery can impact other aspects of a running system and all of the configuration changes performed after the backup was taken will be lost.

To recover the components:

1. Stop the cluster using the WLST stop command:

```
shutdown('cluster_name', 'Cluster')
```

2. Stop all processes, such as the Managed Servers and the Administration Server. For example, to stop the Administration Server on Linux:

```
DOMAIN_HOME/bin/stopWebLogic.sh#username#password#[admin_url]
```

3. Recover the Administration Server configuration by recovering the domain home backup to a temporary location. Then, restore the config directory to the following location:

```
DOMAIN_HOME/config
```

4. Start the Administration Server. For example:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

5. Start the cluster. You can use the Oracle WebLogic Server Administration Console or WLST. For example, to use the WLST start command:

```
start('cluster_name', '#Cluster')
```

The latest configuration is retrieved from the Administration Server to every member of the cluster.

17.2.7 Recovering a Cluster

The following topics describe how to recover a cluster:

- [Recovering a Cluster After Deletion or Cluster-Level Configuration Changes](#)
- [Recovering a Cluster After Membership Is Mistakenly Modified](#)

17.2.7.1 Recovering a Cluster After Deletion or Cluster-Level Configuration Changes

In this scenario, the cluster has been erroneously deleted or the cluster-level configuration, such as the JMS configuration or container-level data sources, was mistakenly changed and committed. The server cannot be started or does not operate

properly or the services running inside the server are not starting. You cannot ascertain what was changed.

Caution: Performing a domain-level recovery can impact other aspects of a running system and all of the configuration changes performed after the backup was taken will be lost.

If the configuration changes are few, then the easiest way is to redo the configuration changes. If that is not feasible, use the following procedure to recover the configuration:

1. Stop the cluster. You can use the Oracle WebLogic Server Administration Console or WLST. For example, to use WLST:

```
shutdown('cluster_name',#'Cluster')
```

2. Stop the Administration Server. For example:

```
DOMAIN_HOME/bin/stopWebLogic.sh#username#password#[admin_url]
```

3. Recover the Administration Server configuration by recovering the domain home backup to a temporary location. Then, restore the config directory to the following location:

```
DOMAIN_HOME/config
```

4. Start the Administration Server. For example:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

5. Start the cluster. You can use the Oracle WebLogic Server Administration Console or WLST. For example, to use WLST:

```
start('cluster_name',#'Cluster')
```

17.2.7.2 Recovering a Cluster After Membership Is Mistakenly Modified

You can recover a cluster when the cluster's membership has been mistakenly modified. For example, if you inadvertently delete a member from the cluster, you can restore the member to the cluster.

Caution: Performing a domain-level recovery can impact other aspects of a running system and all of the configuration changes performed after the backup was taken will be lost.

To recover the cluster membership:

1. Stop all processes, such as the Managed Servers and the Administration Server. For example, to stop the Administration Server on Linux:

```
DOMAIN_HOME/bin/stopWebLogic.sh#username#password#[admin_url]
```

2. Recover the Administration Server configuration by recovering the domain home backup to a temporary location. Then, restore the config directory to the following location:

```
DOMAIN_HOME/config
```

3. Start the Administration Server. For example:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

The deleted member is now back in the cluster.

4. Start all processes, such as the Managed Servers. For example, to start the Managed Server on Linux, use the following script:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

5. Start the cluster. You can use the Oracle WebLogic Server Administration Console or WLST. For example, to use WLST:

```
start('cluster_name',#'Cluster')
```

The deleted member is now part of the cluster.

6. Start all cluster members if they are not started:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

17.2.8 Recovering Applications

The following topics describe how to recover an application:

- [Recovering Application Artifacts](#)
- [Recovering a Redeployed Application That Is No Longer Functional](#)
- [Recovering an Undeployed Application](#)
- [Recovering a Composite Application](#)

Note the following about recovering applications:

- If the application is staged, the Administration server copies the application bits to the staged directories on the Managed Server hosts.
- If the deployment mode is nostage or external_stage, ensure that additional application artifacts are available. For example, applications may reside in directories outside of the domain directory. Make your application files available to the new Administration Server by copying them from backups or by using a shared disk. Your application files should be available in the same relative location on the new file system as on the file system of the original Administration Server.

See Also: *Deploying Applications to Oracle WebLogic Server* for information about deploying applications

17.2.8.1 Recovering Application Artifacts

If an application's artifacts, such as the .ear file, have been lost or corrupted, you can recover the application.

To recover the application:

1. Start the Managed Server to which the application was deployed. For example:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

This synchronizes the configuration with the Administration Server.

On each Managed Server restart, the configuration and application artifacts are retrieved from the Administration Server.

17.2.8.2 Recovering a Redeployed Application That Is No Longer Functional

If a Java EE application was redeployed to a Managed Server (whether or not the Managed Server is part of a cluster) and the application is no longer functional, you can recover it.

To recover the application:

1. Recover the application files from backup, if needed.
2. Redeploy the old version of the application from the backup.

You cannot just copy the original ear file. Even if the original ear file (from the backup) is copied back to the Managed Server stage directory and you restart the Managed Server, the application is still not recovered. You must redeploy the original version.

17.2.8.3 Recovering an Undeployed Application

If a deployed application was undeployed from Oracle WebLogic Server, you can recover it.

To recover the application:

1. Recover the application files from backup, if needed.
2. Redeploy the old version of the application from the backup. If the application was deployed to a cluster, redeploy the application to the same cluster.

You cannot just copy the original ear file. Even if the original ear file (from the backup) is copied back to the Managed Server stage directory and you restart the Managed Server, the application is still not recovered. You must redeploy the original version.

17.2.8.4 Recovering a Composite Application

A new version of a composite application was redeployed to a Managed Server or cluster. The application is no longer functional.

To recover the application:

1. Recover the application files from backup, if needed.
2. Redeploy the old version of the application. If the application was deployed to a cluster, redeploy the application to the same cluster.

17.2.9 Recovering a Database

If your database that contains your metadata repository, including the MDS Repository, is corrupted, you can recover it using RMAN. You can recover the database at the desired granularity, either a full recovery or a tablespace recovery.

For best results, recover the database to the most current state, using point-in-time recovery (if the database is configured in Archive Log Mode.) This ensures that the latest data is recovered. For example:

```
rman>#restore#database;
rman>#recover#database;
```

See *Creating Schemas with the Repository Creation Utility* for the schemas used by each component.

For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*.

17.3 Recovering After Loss of Host

This section describes how to recover your Oracle Fusion Middleware environment after losing the original operating environment. For example, you could have a serious system malfunction or loss of media. The sections includes the following topics:

- [Recovering After Loss of Oracle WebLogic Server Domain Host](#)
- [Recovering After Loss of Standalone Domain Host](#)
- [Recovering After Loss of Administration Server Host](#)
- [Recovering After Loss of Managed Server Host](#)
- [Recovering After Loss of Component Host](#)
- [Additional Actions for Recovering Entities After Loss of Host](#)
- [Recovering After Loss of Database Host](#)

Note: When you are recovering in the case of loss of host, you must restore the files using the same path as on the original host.

17.3.1 Recovering After Loss of Oracle WebLogic Server Domain Host

To recover an Oracle WebLogic Server domain:

1. Change to the directory that you want to be the parent directory of the Domain home directory. Use the same directory structure as in the original environment.

2. Recover the domain directory from backup:

```
(UNIX) #tar#-xf#domain_backup_042013.tar#  
(Windows) #jar#xf#domain_backup_042013.jar#
```

3. Start all relevant processes. That is, start all processes that are related to the domain. For example, start the Administration Server:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

4. If you cannot start the Administration Server, recover it, as described in [Section 17.3.3](#).
5. If you cannot start a Managed Server, recover it, as described in [Section 17.3.4](#).

17.3.2 Recovering After Loss of Standalone Domain Host

If you lose a host that contains a standalone domain, you can recover it to the same host or a different host, as described in the following topics:

- [Recovering a Standalone Domain to the Same Host](#)
- [Recovering a Standalone Domain to a Different Host](#)

17.3.2.1 Recovering a Standalone Domain to the Same Host

In this scenario, you recover the standalone domain to the same host after the operating system has been reinstalled.

To recover the standalone domain to the same host:

1. If Node Manager or a system component, such as Oracle HTTP Server are running, stop them.

2. If it is corrupted, recover the Oracle home:

```
(UNIX) #tar#xf#oracle_home_backup_05_21_2013.tar
(Windows) #jar#xf#oracle_home_backup_05_21_2013.jar
```

3. Recover the domain home:

```
(UNIX) #tar#xf#domain_backup_05_21_2013.tar
(Windows) #jar#xf#domain_backup_05_21_2013.jar
```

4. Start Node Manager:

```
(UNIX) #DOMAIN_HOME/bin/startNodeManager.sh
(Windows) #DOMAIN_HOME\bin\startNodeManager.cmd
```

5. Start any system components, such as Oracle HTTP Server, that are in the domain:

```
(UNIX) #Domain_Home/bin/startComponent.sh#ohs1
(Windows) #Domain_Home\bin\startComponent.cmd#ohs1
```

17.3.2.2 Recovering a Standalone Domain to a Different Host

In this scenario, you recover the standalone domain to a different host.

To recover the standalone domain to a different host:

1. Recover the Oracle home:

```
(UNIX) #tar#xf#oracle_home_backup_05_21_2013.tar
(Windows) #jar#xf#oracle_home_backup_05_21_2013.jar
```

2. Recover the domain home:

```
(UNIX) #tar#xf#domain_backup_05_21_2013.tar
(Windows) #jar#xf#domain_backup_05_21_2013.jar
```

3. In a standalone domain, by default, Node Manager is listening on localhost. However, if it is not, you can update the ListenAddress by using the following WLST commands:

```
readDomain('Domain_Home')
cd('/')
cd('NMPProperties')
set('ListenAddress','localhost')
set('ListenPort',9001)
```

4. Start Node Manager:

```
(UNIX) #DOMAIN_HOME/bin/startNodeManager.sh
(Windows) #DOMAIN_HOME\bin\startNodeManager.cmd
```

5. Update any system component configuration files manually.

See [Section 17.3.5](#) for details for specific components.

6. Start any system components, such as Oracle HTTP Server, that are in the domain. For example:

```
(UNIX) #Domain_Home/bin/startComponent.sh#ohs1
(Windows) #Domain_Home\bin\startComponent.cmd#ohs1
```

7. Update the Oracle Inventory, as described in [Section 17.3.6.4](#).

8. For Windows, update the Windows Registry, as described in [Section 17.3.6.5](#).

17.3.3 Recovering After Loss of Administration Server Host

If you lose a host that contains the Administration Server, you can recover it to the same host or a different host, as described in the following topics:

- [Recovering the Administration Server to the Same Host](#)
- [Recovering the Administration Server to a Different Host](#)

17.3.3.1 Recovering the Administration Server to the Same Host

In this scenario, you recover the Administration Server either to the same host after the operating system has been reinstalled or to a new host that has the same host name. For example, the Administration Server is running on Host A and the Managed Server is running on Host B. Host A has failed for some reason and the Administration Server must be recovered.

To recover the Administration Server to the same host:

1. Recover the file system. For example, recover the domain containing the Administration Server, as described in [Section 17.3.1](#).
2. Attempt to start the Administration Server. For example:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

If the Administration Server starts, you do not need to take any further steps.

3. If the Administration Server fails to start, take the following steps on Host A:
 - a. Stop all relevant processes. That is, stop all processes that are related to the domain, such as the Administration Server and Managed Servers. For example, to stop the Administration Server on Linux:

```
DOMAIN_HOME/bin/stopWebLogic.sh#username#password#[admin_url]
```

- b. Recover the Oracle home, if needed:

```
tar#-xf#oracle_home_backup_042013.tar#
```

- c. If the domain directory does not reside in the Oracle home, recover the domain directory from backup. First change to the directory that you want to be the parent of the Domain home, then:

```
tar#-xf#domain_backup_042013.tar#
```

- d. Start the Administration Server. For example:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

- e. Start the Managed Servers, specifying the Administration URL for the host:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

- f. Start Node Manager:

```
cd#DOMAIN_HOME/bin
./startNodeManager.sh
```

17.3.3.2 Recovering the Administration Server to a Different Host

In this scenario, the Administration Server is running on Host A and the Managed Server is running on Host B. Host A has failed for some reason and the Administration Server must be moved to Host C.

To recover the Administration Server to a different host:

1. Recover the Oracle home to Host C (the new Host):

```
tar#-xf#oracle_home_backup_042013.tar#
```

2. If the domain directory does not reside in the Oracle home, recover the domain directory from backup. First change to the directory that you want to be the parent of the Domain home, then:

```
tar#-xf#domain_backup_042013.tar#
```

3. If the Administration Server has a Listen address, create a new machine with the new host name, as described in [Section 17.3.6.3](#).

4. Start the Administration Server. For example:

```
DOMAIN_HOME/bin/startWebLogic.sh#
```

5. Using WLST, connect to the Administration Server and then enroll Node Manager running in the new host with the Administration Server:

```
connect('username','password','t3://host:port')
nmEnroll('/scratch/oracle/config/domains/domain_name',
##'DOMAIN_HOME/nodemanager')
```

6. Edit the Node Manager properties file, changing the Listen Address property. For a domain-based Node Manager, the file is located at:

```
DOMAIN_HOME/nodemanager/nodemanager.properties
```

Alternatively, you can use the following WLST commands to change the property:

```
readDomain('Domain_Home')
cd('/')
cd('NMProperties')
set('ListenAddress','localhost')
set('ListenPort',port_num)
```

7. Start Node Manager on Host C if it was configured on the original host:

```
cd#DOMAIN_HOME/bin
./startNodeManager.sh
```

8. Start the Managed Servers. The section "Restarting a Failed Administration Server" in the *Administering Server Startup and Shutdown for Oracle WebLogic Server* describes different ways to restart them, depending on how they were configured.

One option is to use the following script, specifying the Administration URL of the new host:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

9. Ensure that additional application artifacts are available. For example, if the deployment mode is `nostage` or `external_stage`, applications may reside in directories outside of the domain directory. Make your application files available to the new Administration Server by copying them from backups or by using a shared disk. Your application files should be available in the same relative location on the new file system as on the file system of the original Administration Server.

If the application is staged, the Administration Server copies the application bits to the staged directories on the Managed Server hosts.

10. Update Oracle Inventory, as described in [Section 17.3.6.4](#).

11. On Windows, recover the Windows Registry, as described in [Section 17.3.6.5](#)
12. If your environment contains Oracle HTTP Server, modify the `mod_wl_ohs.conf` file, as described in [Section 17.3.6.2](#).

Now you can start and stop the Managed Server on Host B using the Administration Console running on Host C.

If you are recovering the Administration Server for a Web Tier installation, see [Section 17.3.6](#) for information about additional actions you must take.

17.3.4 Recovering After Loss of Managed Server Host

If you lose a host that contains a Managed Server, you can recover it to the same host or a different host, as described in the following topics:

- [Recovering a Managed Server to the Same Host](#)
- [Recovering a Managed Server to a Different Host](#)

17.3.4.1 Recovering a Managed Server to the Same Host

In this scenario, you recover a Managed Server to the same host after the operating system has been reinstalled or to a new host that has the same host name. The Administration Server is running on Host A and the Managed Server is running on Host B. Host B failed for some reason and the Managed Server must be recovered to Host B.

To recover a Managed Server to the same host:

1. Start Node Manager on Host B:

```
cd#DOMAIN_HOME/bin
./startNodeManager.sh
```

2. Start the Managed Server. For example:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

If the Managed Server starts, it connects to the Administration Server and updates its configuration changes. You do not need to take any further steps.

3. If the Managed Server fails to start or if the file system is lost, take the following steps:

- a. Recover the Oracle home to Host B from the backup, if required:

```
tar#-xf#oracle_home_backup_042013.tar#
```

- b. Stop Node Manager as described in [Section 4.2.2](#).

- c. Create a domain template jar file for the Administration Server running in Host A, using the pack utility. For example:

```
pack.sh#-domain=/scratch/oracle/config/domains/domain_name
##-template=/scratch/temp.jar#-template_name=test_install#
##-template_author=myname#-log=/scratch/logs/my.log#-managed=true
```

Specifying the `-managed=true` option packs up only the Managed Servers. If you want to pack the entire domain, omit this option.

- d. Unpack the domain template jar file in Host B, using the unpack utility:

```
unpack.sh#-template=/scratch/temp.jar
##-domain=/scratch/oracle/config/domains/domain_name#
```

```
##-log=/scratch/logs/new.log#-log_priority=info
```

- e. Ensure that the application artifacts are accessible from the Managed Server host. That is, if the application artifacts are not on the same server as the Managed Server, they must be in a location accessible by the Managed Server.

Note:

- For applications that are deployed in `nostage` and `external_stage` mode, copy the application artifacts from the Administration Server host directory.
- For applications that are deployed in `stage` mode, the Administration server copies the application bits to the staged directories on the Managed Server hosts.

See *Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

- f. Update the Node Manager property `ListenAddress` by using the following WLST commands:

```
readDomain('Domain_Home')
cd('/')
cd('NMPProperties')
set('ListenAddress','localhost')
set('ListenPort',9001)
```

- g. If Node Manager is not started, start it:

```
cd#DOMAIN_HOME/bin
./startNodeManager.sh
```

- h. Start the Managed Server. For example:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

The Managed Server connects to the Administration Server and updates its configuration changes.

17.3.4.2 Recovering a Managed Server to a Different Host

In this scenario, the Administration Server is running on Host A and the Managed Server is running on Host B. Host B failed for some reason and the Managed Server must be recovered to Host C. There are two machines, which are logical representations of the computer that hosts one or more WebLogic Servers, `machine_1` on Host A and `machine_2` on Host B.

Important: Recover the Oracle home to the same location as the original.

To recover a Managed Server to a different host:

1. Recover the Oracle home for the Managed Server to Host C.

```
tar#-xf#oracle_home_backup_042013.tar#
```

2. Reconfigure the topology to point to the new host:

- a. To avoid an inconsistent backup, do not make any configuration changes until the backup is completed. To ensure that no changes are made in the WebLogic Server domain, lock the WebLogic Server configuration, as described in [Section 2.3.2](#).

- b. In the WebLogic Server Administration Console, change the configuration of machine_2, to point it to the new host:

From the left pane of the Console, expand **Environment** and then select **Machines**. Select machine_2 and select the Configuration tab. Then select the Node Manager tab. Change the **Listen Address** to the address for Host C. Click **Save**.

If you identify the Listen Address by IP address, you must disable Host Name Verification on the Administration Servers that access Node Manager. For more information and instructions, see "Using Hostname Verification" in *Administering Security for Oracle WebLogic Server*.

- c. Change the Managed Server configuration to point to the new host:

From the left pane of the Console, expand **Environment** and then **Servers**. Then, select the name of the server. Select the **Configuration** tab, then the **General** tab.

Change the **Machine** to machine_2.

Change **Listen Address** to the new host. (If the listening address was set to blank, you do not need to change it.)

- d. Unlock the Oracle WebLogic Server configuration by clicking **Release Configuration** on the WebLogic Server Administration Console,
3. Take any additional steps needed for components as described in [Table 17-1](#).
 4. Stop Node Manager as described in [Section 4.2.2](#).
 5. Create a domain template jar file from the Administration Server running in Host A, using the pack utility. For example:

```
pack.sh#-domain=/scratch/oracle/config/domains/domain_name
##-template=/scratch/temp.jar#-template_name=test_install#
##-template_author=myname#-log=/scratch/logs/my.log#-managed=true
```

Specifying the -managed=true option packs up only the Managed Servers. If you want to pack the entire domain, omit this option.

6. Unpack the domain template jar file on Host C, using the unpack utility:

```
unpack.sh#-template=/scratch/temp.jar
###-domain=/scratch/oracle/config/domains/domain_name
###-log=/scratch/logs/new.log#-log_priority=info
```

7. Ensure that the application artifacts are accessible from the Managed Server host. That is, if the application artifacts are not on the same server as the Managed Server, they must be in a location accessible by the Managed Server.

Note:

- For applications that are deployed in `nostage` and `external_stage` mode, copy the application artifacts from the Administration Server host directory.
- For applications that are deployed in `stage` mode, the Administration server copies the application bits to the staged directories on the Managed Server hosts.

See *Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

8. Update the `ListenAddress` by using the following WLST commands:

```
readDomain('Domain_Home')
cd('/')
cd('NMPProperties')
set('ListenAddress','localhost')
set('ListenPort',9001)
```

9. Start Node Manager on Host C, if it is not started:

```
cd#DOMAIN_HOME/bin
./startNodeManager.sh
```

10. Start the Managed Server. For example:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh#managed_server_name#admin_url#
```

The Managed Server connects to the Administration Server and updates its configuration changes.

11. Update Oracle Inventory, as described in [Section 17.3.6.4](#).
12. On Windows, recover the Windows Registry, as described in [Section 17.3.6.5](#)
13. If your environment contains Oracle HTTP Server, modify the `mod_wl_ohs.conf` file, as described in [Section 17.3.6.2](#).

Now you can start and stop the Managed Server on Host C using the Administration Server running on Host A.

17.3.5 Recovering After Loss of Component Host

If you lose a host that contains a component (and its Managed Server, if applicable), you can recover most components to the same host or a different host using the procedures described in the following topics:

- [Recovering a Java Component to the Same Host](#)
- [Recovering a Java Component to a Different Host](#)
- [Recovering a System Component to the Same Host](#)
- [Recovering a System Component to a Different Host](#)

Some components require additional actions, which are described in the sections listed in [Table 17-1](#).

Table 17–1 Recovery Procedures for Loss of Host for Particular Components

Component	Procedure
Oracle Data Integrator	Section 17.3.5.6
Oracle HTTP Server	Section 17.3.5.5.2

17.3.5.1 Recovering a Java Component to the Same Host

To recover a Java component to the same host:

1. Recover the Managed Server, as described in [Section 17.3.4.1](#).
2. If the component requires additional steps, as noted in [Table 17–1](#), take those steps.

17.3.5.2 Recovering a Java Component to a Different Host

To recover a Java component to a different host, recover the Managed Server, as described in [Section 17.3.4.2](#).

However, note that some components require additional steps, as noted in [Table 17–1](#).

17.3.5.3 Recovering a System Component to the Same Host

To recover a system component, such as Oracle HTTP Server, to the same host:

- For system components, such as Oracle HTTP Server, in a standalone domain, you recover the domain, as described in [Section 17.3.2](#).
- For system components, such as Oracle HTTP Server in an Oracle WebLogic Server domain, you recover the domain, as described in [Section 17.3.1](#).

However, note that some components require additional steps, as noted in [Table 17–1](#).

17.3.5.4 Recovering a System Component to a Different Host

To recover a system component, such as Oracle HTTP Server, to a different host:

- For system components in a standalone domain, you recover the domain, as described in [Section 17.3.2](#).
- For system components in an Oracle WebLogic Server domain, you recover the domain, as described in [Section 17.3.1](#).

However, note that most components require additional steps, as noted in [Table 17–1](#).

17.3.5.5 Recovering Web Tier Components to a Different Host

The Web tier consists of Oracle HTTP Server. The following topics describe how to recover these components to a different host:

- [Recovering Oracle HTTP Server in a Standalone Domain to a Different Host](#)
- [Recovering Oracle HTTP Server in a WebLogic Server Domain to a Different Host](#)

17.3.5.5.1 Recovering Oracle HTTP Server in a Standalone Domain to a Different Host

To recover Oracle HTTP Server in a standalone domain:

1. Follow steps 1 through 4 in [Section 17.3.2](#).
2. Update the configuration files in the following directory:

```
(UNIX) #DOMAIN_HOME/config/fmwconfig/components/OHS/instance_name
(Windows) #DOMAIN_HOME\config\fmwconfig\components\OHS\instance_name
```

For example, update the IP Address and host name in `httpd.conf`, `admin.conf`, and `mod_wl_ohs.conf` (if required).

3. Follow steps 6 through 8 in [Section 17.3.2](#).

17.3.5.5.2 Recovering Oracle HTTP Server in a WebLogic Server Domain to a Different Host To recover Oracle HTTP Server in an Oracle WebLogic Server domain to a different host:

1. Recover the domain, as described in [Section 17.3.1](#).
2. Change the configuration of the Oracle HTTP Server instance that was on Host B:
 - a. In Fusion Middleware Control, from the navigation pane, expand the domain, then **Web Tier**.
 - b. Select the Oracle HTTP Server instance, such as `ohs1`.
 - c. From the Oracle HTTP Server menu, select **Administration**, then **Ports Configuration**.
 - d. For each port in the table, select the port, then click **Edit**. Change the **IP Address**.
Note that if ANY is selected, you do not need to make any changes.
 - e. Click **OK**.
3. Update the `mod_wl_ohs` wiring for each Oracle HTTP Server instance:
 - a. In Fusion Middleware Control, from the navigation pane, expand the domain, then **Web Tier**.
 - b. Select the Oracle HTTP Server instance, such as `ohs1`.
 - c. From the Oracle HTTP Server menu, select **Administration**, then **mod_wl_ohs Configuration**.
 - d. In the Locations section, click **AutoFill**.
All valid WebLogic Server endpoint locations are displayed.
 - e. Click **Apply**.
4. Restart any Oracle HTTP Server instances that are not on the failed machine by navigating to that instance and clicking **Start Up**.
5. Start the Oracle HTTP Server instances on Host C by navigating to that instance and clicking **Start Up**.

17.3.5.6 Recovering Oracle Data Integrator to a Different Host

To recover Oracle Data Integrator, following the procedures in one or both of these topics, depending on the failure:

- [Recovering Oracle Data Integrator Repository](#)
- [Recovering Oracle Data Integrator Agents to a Different Host](#)

17.3.5.6.1 Recovering Oracle Data Integrator Repository If the Oracle Data Integrator Repository must be restored to a different host:

1. Restore the database, as described in [Section 17.3.7](#).
2. Connect to the restored Oracle Data Integrator repository using ODI Studio. Create a new connection for the master repository to the new database host, as described in "Connecting to the Master Repository" in the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

3. Edit each of the Work Repositories. Click **Connection** and edit the connection information so that the JDBC URL points to the new database host containing the work repository.
4. For the Oracle Data Integrator JEE Agent repository configuration, in the Oracle WebLogic Server configuration, edit the data sources to match the new database host location.
5. Update the Oracle Data Integrator Standalone Agent Repository configuration using the following WLST offline commands:

```
cd#ORACLE_HOME/odi/common/bin
./wlst.sh
readDomain('DOMAIN_DIRECTORY')
cd('/JdbcSystemResource/OdiMasterRepository/JdbcResource/OdiMasterRepository/JDBCDataSourceParams/NO_NAME_0');
set('URL','NEW_JDBC_URL_TO_RECOVERED_DB');
updateDomain();
exit();
```

17.3.5.6.2 Recovering Oracle Data Integrator Agents to a Different Host

To recover Oracle Data Integrator agents to a different host:

1. Restore Oracle Data Integrator JEE Agent by restoring the Managed Server, as described in [Section 17.3.4](#).
2. Restore the Oracle Data Integrator Standalone system component, as described in [Section 17.3.5](#)
3. Use ODI Studio to edit each physical agent's configuration and provide the updated Host Name value and, if changed, the Port value.
4. Update Oracle Data Integrator Standalone Agent's host and port configuration using the following commands:

```
cd#ORACLE_HOME/odi/common/bin
./wlst.sh
readDomain('DOMAIN_HOME')
cd('ODI/ODI_STANDALONE_AGENT_NAME')
set('ServerListenAddress','UPDATED_HOST_NAME');
set('ServerListenPort','UPDATED_PORT');
updateDomain();
exit();
```

5. Restart the standalone agents and the Oracle Data Integrator applications deployed in Oracle WebLogic Server.

17.3.6 Additional Actions for Recovering Entities After Loss of Host

Depending on the entity that you are recovering, you may need to take additional actions after loss of host. The sections about each entity may require you to follow one or more of the following procedures. If so, that is noted in the section describing how to recover the entity.

The following topics describe the actions you may need to take:

- [Recovering Fusion Middleware Control to a Different Host](#)
- [Modifying the mod_wl_ohs.conf File](#)
- [Creating a New Machine for Certain Components](#)
- [Updating Oracle Inventory](#)

- [Recovering the Windows Registry](#)

17.3.6.1 Recovering Fusion Middleware Control to a Different Host

To recover Fusion Middleware Control to a different host, update properties using the System MBean Browser:

1. In Fusion Middleware Control, from the WebLogic Domain menu, select **System MBean Browser**.
2. In the System MBean Browser pane, expand **Application-Defined MBeans**, then **emoms.props**, then **Server: AdminServer**, then **Application: em**, and then **Properties**.
3. Click **emoms.properties**.
4. In the Attributes pane, select the Operations tab and click **setProperty**.
5. Change the value of the following properties to the new host name:
 - oracle.sysman.emSDK.svlt.ConsoleServerHost
 - oracle.sysman.emSDK.svlt.ConsoleServerName

For example, for Key, enter oracle.sysman.emSDK.svlt.ConsoleServerHost. Then, for value, enter host.example.com:7001_Management_Service.
6. Click **Invoke**.

17.3.6.2 Modifying the mod_wl_ohs.conf File

When you recover an Administration Server or a Managed Server to a different host and your environment includes Oracle HTTP Server, you must modify the following file on the new host:

```
(UNIX) #DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_name/mod_wl_ohs.conf
(Windows) #DOMAIN_HOME\config\fmwconfig\components\OHS\ohs_name\mod_wl_ohs.conf
```

Note that with Oracle HTTP Server in a WebLogic Server domain, this directory is in the Domain home of the Administration Server. With Oracle HTTP Server in a standalone domain, this directory is the Domain home of Oracle HTTP Server.

Modify all of the instances of the host name, port, and clusters (elements such as WebLogicHost, WebLogicPort, and WebLogicCluster) entries in that file. For example:

```
<Location#/console>
####SetHandler#weblogic-handler
####WebLogicHost#Admin_Host
####WebLogicPort#Admin_Port
####WLProxySSL#ON
####WLProxySSLPassThrough#ON
</Location>
```

17.3.6.3 Creating a New Machine for Certain Components

If the Administration Server has a Listen address, you must create a new machine with the new host name before you start the Administration Server:

Take the following steps:

1. Create a new machine with the new host name. Use the following WLST commands, in offline mode:

```
wls:/offline>#readDomain('DOMAIN_HOME')
wls:/offline/sampledmain>#machine#=#create('newhostname',#'Machine')
```

```
wls:/offline/sampledomain>#cd('/Machine/newhostname')
wls:/offline/sampledomain>#nm#=#create('newhostname',#'NodeManager')
wls:/offline/sampledomain>#cd('/Machine/newhostname/NodeManager/newhostname')
wls:/offline/sampledomain>#set('ListenAddress',#'newhostname')
wls:/offline/sampledomain>#updateDomain()
wls:/offline/sampledomain>#exit()
```

2. For the Administration Server, set the machine with the new host name, using the following WLST command, in offline mode:

```
wls:/offline>#readDomain('DOMAIN_HOME')
wls:/offline/sampledomain>#cd#('/Machine/newhostname')
wls:/offline/sampledomain>#machine#=#cmo
wls:/offline/sampledomain>#cd#('/Server/AdminServer')
wls:/offline/sampledomain>#set('Machine',#machine)
wls:/offline/sampledomain>#updateDomain()
wls:/offline/sampledomain>#exit()
```

3. Set the listen port for the Administration Server:

```
wls:/offline/sampledomain>#readDomain('DOMAIN_HOME')
wls:/offline/sampledomain>#cd('/Server/AdminServer')
wls:/offline/sampledomain>#cmo.setListenPort(8001)
wls:/offline/sampledomain>#updateDomain()
wls:/offline/sampledomain>#exit()
```

4. If required, update the Administration Server listen address, using WLST:

```
readDomain('DOMAIN_HOME')
cd('/Server/AdminServer')
cmo.getListenAddress()
cmo.setListenAddress('newhostname')
updateDomain()
exit()
```

17.3.6.4 Updating Oracle Inventory

For many components, when you recover to a different host, as in the case of loss of host, you must update the Oracle inventory. To do so, execute the following script:

```
(UNIX)#ORACLE_HOME/oui/bin/attachHome.sh
(Windows)#ORACLE_HOME\oui\bin\attachHome.cmd
```

17.3.6.5 Recovering the Windows Registry

When you recover any component to a different host on Windows, as in the case of loss of host, you must import any Windows Registry keys related to Oracle Fusion Middleware to the new host. (You exported the Registry keys in [Section 16.3.3](#).)

Recover the following Registry key.

```
HKEY_LOCAL_MACHINE\Software\Oracle
```

In addition, recover each node that begins with **Oracle** within the following registry keys:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services
```

To import a key that you have previously exported, use the following command:

```
regedit##/I##FileName
```

For example:

```
regedit#/I#C:\oracleregistry.reg#
```

You can also use the Registry Editor to import the key. See the Registry Editor Help for more information.

17.3.7 Recovering After Loss of Database Host

If the host that contained your database is lost, you can recover the database using RMAN.

For example:

```
rman>#restore#database;  
rman>#recover#database;
```

For best results, recover the database to the most current state, using point-in-time recovery (if the database is configured in Archive Log Mode.) This ensures that the latest data is recovered. Also, use the same name for the database. Note the following:

See *Creating Schemas with the Repository Creation Utility* for the schemas used by each component.

For detailed steps about recovering a database and using RMAN, see the *Oracle Database Backup and Recovery User's Guide*.

Part VIII

Advanced Administration: Expanding Your Environment

This part describes how to expand your Oracle Fusion Middleware environment.

It contains the following chapters:

- [Chapter 18, "Scaling Up Your Environment"](#)
- [Chapter 19, "Moving from a Test to a Production Environment"](#)

Scaling Up Your Environment

You can expand your environment by adding Managed Servers, expanding your domain to include other products, creating a cluster of Managed Servers, copying existing Oracle homes or existing Oracle Fusion Middleware components such as Oracle HTTP Server, as described by the following sections:

- [Overview of Scaling Up Your Environment](#)
- [Extending a Domain to Support Additional Components](#)
- [Adding Additional Managed Servers to a Domain](#)
- [Creating Clusters](#)
- [Copying an Oracle Home or Component](#)

18.1 Overview of Scaling Up Your Environment

Scalability is the ability of a system to provide throughput in proportion to, and limited only by, available hardware resources. A scalable system is one that can handle increasing numbers of requests without adversely affecting response time and throughput.

The growth of computational power within one operating environment is called vertical scaling. Horizontal scaling is leveraging multiple systems to work together on a common problem in parallel.

Oracle Fusion Middleware scales both vertically and horizontally.

Oracle Fusion Middleware provides great vertical scalability, allowing you to add more Managed Servers or components to the same host. This is known as scale up.

Horizontally, Oracle Fusion Middleware can provide failover capabilities to another host computer. That way, if one computer goes down, your environment can continue to serve the consumers of your deployed applications. This is also known as scaling out or machine scale out. For information about scaling out, see "Scaling Out a Topology" in the *High Availability Guide*.

Deploying a high availability system minimizes the time when the system is down (unavailable) and maximizes the time when it is running (available). Oracle Fusion Middleware is designed to provide a wide variety of high availability solutions, ranging from load balancing and basic clustering to providing maximum system availability during catastrophic hardware and software failures.

High availability solutions can be divided into two basic categories: local high availability and disaster recovery.

See Also:

- *High Availability Guide* for more information about high availability
- *Oracle Fusion Middleware Disaster Recovery Guide*

18.2 Extending a Domain to Support Additional Components

When you create an Oracle WebLogic Server domain, you create it using a particular domain template. That template supports a particular component or group of components, such as Oracle WebLogic Server. If you want to add other components, such as Oracle HTTP Server, to that domain, you can extend the domain by creating additional Managed Servers in the domain, using a domain template for the component which you want to add.

When you extend a domain, the domain must be offline.

To extend a domain, you use the Oracle WebLogic Server Configuration Wizard from an Oracle home into which the desired component has been installed. Then, you select the domain that you want to extend and the component you want to add. For detailed information, see "Configuring Your WebLogic Domain" in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

For example, to extend a domain that initially was created to support Oracle Application Development Framework so that it can now also support Oracle HTTP Server:

1. Use RCU to add any required schemas for the component, as described in *Creating Schemas with the Repository Creation Utility*.
2. Install Oracle HTTP Server, as described in *Installing and Configuring Oracle HTTP Server*.
3. From the Oracle home, invoke the Configuration Wizard, using the following command:

```
(UNIX) #ORACLE_HOME/oracle_common/common/bin/config.sh  
(Windows) #ORACLE_HOME\oracle_common\common\bin\config.cmd
```

The Configuration Wizard's Welcome screen is displayed.

4. Select **Update an existing domain**.
5. In Domain Location, specify the location of the domain.
6. Click **Next**.
7. Select **Update Domain Using Product Templates**.
8. Select **Oracle HTTP Server (colocated)**.
9. Click **Next**.
10. Select **Extend my domain automatically to support the following added products**, Then, select the source from which this domain is to be extended. For example, select **Oracle HTTP Server**.
11. Click **Next**.
12. Select either **RCU Data** or **Manual Configuration**. If you select RCU Data, the information is automatically populated. If you select Manual Configuration, click **Next**.

Select the schemas for the new component you added, entering the following information:

- For **Vendor**, select **Oracle**.
- For **Driver**, select **Oracle's Driver (Thin) for Service connections; Versions:9.0.1,9.2.0,10,11**.
- For **Schema Owner**, do not enter anything. Each data source uses the user name specified in the table.
- If you used the same password when you created the schemas, select all of the schemas and enter the password in **Schema Password**.
Alternatively, you can specify different passwords for each data source by selecting each schema individually and entering the password.
- With all of the schemas selected, for **DBMS/Service**, enter the SID of the database.
- With all of the schemas selected, for **Host Name**, enter the host name of the database.
- With all of the schemas selected, for **Port**, enter the listening port of the database.

13. Click Next.

The JDBC Component Schema Test screen is displayed.

14. If the test succeeds, click Next.

The Advanced Configuration screen is displayed.

15. Select System Components.

16. Click Next.

17. Click Add to create a new Oracle HTTP Server instance.

18. Enter a name for the instance and select OHS as the component type.

19. Click Next.

20. The fields in the OHS Server page are prepopulated.

21. Click Next.

22. If you do not want to create a new machine, in the Machines page, click Next.

23. In the Assign System Components page, double-click the server to move it under the machine.

24. Review the information on the screen and if it is correct, click Update.

25. When the operation completes, click Done.

18.3 Adding Additional Managed Servers to a Domain

You can add Managed Servers to a domain to increase the capacity of your system. The Managed Servers can be added to a cluster.

When a Managed Server is added to a cluster, it inherits the applications and services that are targeted to the cluster. When a Managed Server is not added as a part of a cluster, it does not automatically inherit the applications and services from the template.

To add a Managed Server to a domain, you can use Fusion Middleware Control, the Oracle WebLogic Server Administration Console, or WLST.

See: *Oracle WebLogic Server Administration Console Online Help* and *WLST Command Reference for WebLogic Server* for complete information about adding Managed Servers.

To add a Managed Server to a domain using the Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Administration**, then, **Create/Delete Components**.

The Create Components page is displayed.

2. Click **Create**, and select **WebLogic Server**.

The Create WebLogic Server page is displayed.

3. For **Name**, enter a name for the server.

Each server within a domain must have a name that is unique for all configuration objects in the domain. Within a domain, each server, computer, cluster, JDBC connection pool, virtual host, and any other resource type must be named uniquely and must not use the same name as the domain.

4. For **Listen Port**, enter the port number from which you want to access the server instance.

If you run multiple server instances on a single computer, each server must use its own listen port.

5. Select either **Create new WebLogic Machine** or **Select Existing WebLogic Machine**. If you select **Select Existing WebLogic Machine**, select a machine from the table.

6. Specify whether this server is to be a standalone server or a member of an existing cluster:

- If this server is to be part of an existing cluster, select the cluster.
- If this server is to be a standalone server, do not select a cluster.

7. Click **Create**

8. If the server or cluster did not have Oracle JRF applied, apply JRF, as described in [Section 18.3.1](#).

18.3.1 Applying Oracle JRF Template to a Managed Server or Cluster

Oracle JRF (Java Required Files) consists of those components not included in the Oracle WebLogic Server installation and that provide common functionality for Oracle business applications and application frameworks.

JRF consists of several independently developed libraries and applications that are deployed into a common location. The components that are considered part of Java Required Files include Oracle Application Development Framework shared libraries and ODL logging handlers.

You must apply the JRF template to a Managed Server or cluster in certain circumstances. You can only apply JRF to Managed Servers that are in a domain in which JRF was configured. That is, you must have selected Oracle JRF in the Configuration Wizard when you created or extended the domain.

Note the following points about applying JRF:

- When you add a Managed Server to an existing cluster that is already configured with JRF, you do not need to apply JRF to the Managed Server.
- When you add a Managed Server to a domain and the Managed Server requires JRF services, but the Managed Server is not part of a cluster, you must apply JRF to the Managed Server.
- When you create a new cluster and the cluster requires JRF, you must apply JRF to the cluster.
- You do not need to apply JRF to Managed Servers that are added by product templates during the template extension process (though you must select JRF in the Configuration Wizard).
- You must restart the server or cluster after you apply JRF.

Note that if you start the server or cluster using Node Manager (for example, through the Administration Console, which uses Node Manager), you must set the Node Manager property `startScriptEnabled` to `true`. For more information, see [Section 2.8.1](#).

- If you create a server using Fusion Middleware Control, the JRF template is automatically applied.

You use the custom WLST command `applyJRF` to configure the Managed Servers or cluster with JRF. To use the custom WLST commands, you must invoke the WLST script from the Oracle Common home. See [Section 2.4.2](#) for more information.

The format of the `applyJRF` command is:

```
applyJRF(target={server_name#|#cluster_name#|#*},#domainDir=domain_path,
#####[shouldUpdateDomain=#{true#|#false}])
```

You can use the `applyJRF` command online or offline:

- In online mode, the JRF changes are implicitly activated if you use the `shouldUpdateDomain` option with the value `true` (which is the default.) In online mode, this option calls the online WLST `save()` and `activate()` commands.
- In offline mode, you must restart the Administration Server and the Managed Servers or cluster. (In offline mode, if you specify the `shouldUpdateDomain` option with the value `true`, this option calls the WLST `updateDomain()` command.)

For example, to configure the Managed Server `server1` with JRF, use the following command:

```
applyJRF(target='server1',#domainDir='DOMAIN_HOME')
```

To configure all Managed servers in the domain with JRF, specify an asterisk (*) as the value of the `target` option.

To configure a cluster with JRF, use the following command:

```
applyJRF(target='cluster1',#domainDir='DOMAIN_HOME')
```

See Also:

- "Java Required Files Custom WLST Commands" in the *WLST Command Reference for Infrastructure Components*
- [Section H.2.2](#) to use a different version of Spring than that which is supplied with JRF

18.4 Creating Clusters

A WebLogic Server **cluster** consists of multiple WebLogic Server server instances running simultaneously and working together to provide increased scalability and reliability. A cluster appears to clients to be a single WebLogic Server instance. The server instances that constitute a cluster can run on the same computer, or be located on different computers. You can increase a cluster's capacity by adding additional server instances to the cluster on an existing computer, or you can add computers to the cluster to host the incremental server instances. Each server instance in a cluster must run the same version of WebLogic Server.

You can create a cluster of Managed Servers using WLST, the Oracle WebLogic Server Administration Console, or Fusion Middleware Control. This section describes how to create a cluster using Fusion Middleware Control.

To create a cluster of two Managed Servers, `wls_server1` and `wls_server2`, take the following steps:

1. From the WebLogic Domain menu, choose **Administration**, then, **Create/Delete Components**.
The Fusion Middleware Components page is displayed.
 2. Choose **Create**, then **WebLogic Cluster**.
The Create WebLogic Cluster page is displayed.
 3. For **Name**, enter a name for the cluster.
 4. In the Cluster Messaging Mode section, select one of the following:
 - **Unicast**. Then, for **Unicast Broadcast Channel**, enter a channel. This channel is used to transmit messages within the cluster.
 - **Multicast**. Then, for **Multicast Broadcast Channel**, enter a channel. A multicast address is an IP address in the range from 224.0.0.0 to 239.255.255.255. For **Multicast Port**, enter a port number.
-
-
- Note:** You must ensure that the multicast address is not in use.
-
-
5. In the Servers section, select one or more servers to be added to the cluster. In this scenario, select `wls_server1` and `wls_server2`.
 6. Click **Create**.

Now, you have a cluster with two members, `wls_server1` and `wls_server2`.

See Also: *Administering Clusters for Oracle WebLogic Server* for more information about clusters

18.5 Copying an Oracle Home or Component

You can copy an Oracle home or many Oracle Fusion Middleware components to a different location while preserving its configuration. Some situations in which copying Oracle Fusion Middleware is useful are:

- Creating an Oracle home that is a copy of a production, test, or development environment, enabling you to create a new Oracle home or component with all patches applied to it in a single step. This is in contrast to separately installing, configuring and applying any patches to separate components.

- Preparing a "gold" image of a patched home and deploying it to many hosts.

For information about the procedures you use to copy an Oracle home or component, see [Chapter 19](#).

[[Moving from a Test to a Production Environment]]

[This chapter describes how to move Oracle Fusion Middleware from a source environment, such as a test environment, to a target environment, such as a production environment. You can develop and test applications in a source environment, and then eventually roll out the test applications and, optionally, test data to your target environment. You can also use this approach for testing and rolling out upgrades.]

[This chapter includes the following sections:]

- [[[[Introduction to Moving Oracle Fusion Middleware Components]]]]
- [[[Planning for Moving Your Environment]]]
- [[[Common Procedures for Moving to a Target Environment]]]
- [[[Additional Steps for Certain Components]]]
- [[[Incrementally Moving Artifacts]]]
- [[[Moving Distributed Topologies]]]
- [[[Recovering from Test to Production Errors]]]]]

"

[Note]: []

- [[The procedures in this chapter are valid for Oracle Fusion Middleware 12c (12.1.2) and the components that are part of that release.]]
- [[The procedures in this chapter for the most part assume that you are using the standard installation topology, which consists of a WebLogic Server domain that contains an Administration Server and a cluster containing two Managed Servers or a standalone domain.]

[For more information about the standard topology, see ["Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology"] in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.]]]

19.1 [[Introduction to Moving Oracle Fusion Middleware Components]]

[You can move Oracle Fusion Middleware components from a source environment to a target environment.]

[Moving Oracle Fusion Middleware components minimizes the amount of work that would otherwise be required to reapply all the customization and configuration changes made in one environment to another. You can install, configure, customize, and validate Oracle Fusion Middleware in a source environment. Once the system is stable and performs as desired, you can create the target environment by moving a copy of the components and their configurations from the source environment, instead of redoing all the changes that were incorporated into the source environment.]

19.2 [[Planning for Moving Your Environment]

[This section describes important information that you should know before you begin moving your environment. It includes the following topics:]

- [\[\[\[Introduction to Moving Oracle Fusion Middleware Components\]\]\]](#)
- [\[\[\[Checking the Source Environment\]\]\]](#)
- [\[\[\[Preparing the Target Environment\]\]\]](#)
- [\[\[\[Limitations in Moving from Source to Target\]\]\]](#)
- [\[\[\[Overview of Procedures for Moving from a Source to a Target Environment\]\]\]](#)

19.2.1 [[Introduction to the Movement Scripts]

[Oracle Fusion Middleware provides a series of scripts that you can use to move your environment:]

- [\[\[copyBinary: Copies the binary files of the source Oracle home.\]\]](#)
- [\[\[pasteBinary: Applies the copied Oracle home to the target.\]\]](#)
- [\[\[copyConfig: Used for any of the following:\]\]](#)
 - [\[\[\[Copies the configuration of a WebLogic Server domain, including any Java components or system components in the domain.\]\]\]](#)
 - [\[\[Copies the configuration of a standalone domain, including any system components in the domain.\]\]\]](#)
 - [\[\[Copies the configuration of Node Manager.\]\]\]](#)
- [\[\[extractMovePlan: Extracts a move plan from the domain or component.\]\]](#)
- [\[\[pasteConfig: Used for any of the following: \]\]](#)
 - [\[\[\[Applies the copied configuration of a WebLogic Server domain, including any Java components or system components in the domain.\]\]\]](#)
 - [\[\[Applies the copied configuration of a standalone domain, including any system components in the domain.\]\]\]](#)
 - [\[\[Applies the copied configuration of Node Manager.\]\]\]](#)

[The scripts enable you to copy an Oracle home and Oracle WebLogic Server domains, as well as the configuration of certain Oracle Fusion Middleware components, such as Oracle HTTP Server.]

[\[\[Table 19–1\]](#) shows which Oracle Fusion Middleware components support the movement scripts.]

Table 19–1 []Support for Movement Scripts

[Component]	[Supported?]
[Oracle Application Development Framework]	[Yes]
[Oracle Coherence]	[Yes]
[Oracle Data Integrator]	[Yes]
[Oracle HTTP Server]	[Yes]
[Oracle HTTP Server WebGate]	[Yes]
[Oracle Platform Security Services]	[Yes]
[Oracle User Messaging Service]	[Yes]
[Oracle Web Services Manager]	[Yes]
[Oracle WebLogic Server]	[Yes]

19.2.2 [[Checking the Source Environment]

[The procedures in this chapter assume that you have installed and configured Oracle Fusion Middleware on the source environment, including some or all of the following:]

- [[[[Installed one or more databases to be used by Oracle Fusion Middleware components.]]]
- [[Created the needed schemas in the source environment using RCU. See *[[Creating Schemas with the Repository Creation Utility]]*.]]]
- [[Installed and configured Oracle Fusion Middleware products. For example, you have installed Oracle WebLogic Server and Oracle Web Services Manager, created the Oracle home, and configured an Oracle WebLogic Server domain.]

[When you configure the domain, you can choose one of two modes:]

- [[[[Development mode: In this mode, the security configuration is relatively relaxed. User name and password are required to deploy applications.]]]
- [[Production mode: In this mode, the security configuration is relatively stringent, requiring a user name and password to deploy applications and to start the Administration Server.]]]]
- [[Alternatively, you have installed and configured system components, such as Oracle HTTP Server, in a standalone domain.]]]
- [[Configured security policies.]]]
- [[For Oracle Platform Security, created security policies and stored credentials in the Credential Store Framework (CSF).]]]
- [[Deployed one or more applications. The applications may have internal and external references.]]]
- [[Before you execute the copyConfig script in a WebLogic Server domain, make sure that the Administration Server and Managed Servers are running.]]]
- [[For Oracle Web Services Manager, before you execute the copyConfig script, the server on which the Oracle Web Services Manager Policy Manager application is deployed must be running.]]]]]

19.2.3 [[Preparing the Target Environment]

[To use the procedures in this chapter, your target environment must meet the following prerequisites:]

- [[You must use the cloningclient.jar file and movement scripts that are compatible with the version of the Oracle home and components that you want to copy. The procedures in this chapter presume that you are using the current version of the cloningclient.jar file and movement scripts.]]
- [[The target environment must be on the same operating system as the source environment. Also, the operating system architecture must be the same in both environments. For example, both environments must be running 32-bit operating systems or 64-bit operating systems.]

[When you execute the scripts, you must specify a matching Java home. That is, if the Oracle homes are 64 bit, you must specify a 64-bit Java home. If the Oracle homes are 32 bit, you must specify a 32-bit Java home.]]

- [[The target environment must have the same superuser or administrative user as the user at the source environment. After you complete the movement of the installation, you can modify the user on the target environment.]]
- [[The database in the target environment must be the same type of database as in the source environment. For example, if the database in the source environment is an Oracle Database, the database in the target environment must be an Oracle Database. The database on the target environment should be the same version as on the source environment.]]
- [[If the database is not tuned correctly, the copyConfig and pasteConfig operations can incur performance issues. To avoid these performance issues, in addition to following standard database performance tuning guidelines, ensure that you have sufficient RAM allocated for your database for the import of the MDS tables. Also run statistics against the target database by executing the following procedure:]

```
[ [BEGIN]
[dbms_stats.gather_schema_stats(ownname#=>#'[prefix]_MDS', #)
[#####METHOD_OPT#=>#'FORALL#COLUMNS#SIZE#AUTO', ]
[#####CASCADE#=>#TRUE, #ESTIMATE_PERCENT#=>#NULL); ]
[END; ]
[]]
```

[In the procedure, [prefix][_MDS] is the MDS schema name for your installation.]]

- [[The host must have JDK 1.7.0_15 or higher installed. In addition, ensure that the PATH, CLASSPATH, and JAVA_HOME environment variables point to the JDK.]

[For example, the CLASSPATH variable should contain the following:]

```
[[$ORACLE_HOME/lib:/scratch/jdk[n]/lib]
[]]]
```

- [[If you are applying the archive of an Oracle home on a host that does not yet have Oracle Fusion Middleware installed, note the following:]
 - [[Copy the pasteBinary script from the following location in the source host to the target host:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/pasteBinary.sh]
[(Windows)#[ORACLE_COMMON_HOME]\bin\pasteBinary.cmd]
[]]]
```

- [[Copy the following file from the following location in the source host to the target host:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/jlib/cloningclient.jar]
```

- ```
[(Windows)#[ORACLE_COMMON_HOME]\jlib\cloningclient.jar]
[]]]
```
- [[If you run the pasteBinary script from a different location than [ORACLE\_COMMON\_HOME]/[bin], then the pasteBinary script and the cloningclient.jar file must be in the same directory.]
 

[If you are running pasteBinary on a host that has no prior Oracle Fusion Middleware installations, [ORACLE\_COMMON\_HOME]/[bin] will not exist prior to running pasteBinary, and therefore the pasteBinary script and cloningclient.jar must be in the same directory.]]
  - [[Ensure that the files have execute permission.]]]]
  - [[When you execute the copyConfig command on the source environment, any system components can be started or shut down. In either case, the copyConfig operation will complete. This pertains to both WebLogic Server domains and standalone domains.]]
  - [[On Windows, the file MSVCR90.DLL must exist on the target host. Otherwise, pasteConfig will fail. ]
 

[The file is located in:]

```
[[(Windows#32#bit) #C:\Windows\System32#]
[(Windows#64#bit) #C:\Windows\winsxs]]]]]
```

## 19.2.4 [[Limitations in Moving from Source to Target]

[Note the following limitations and restrictions:]

- [[The source and the target environment must use the same encoding. For example, if the source environment uses the encoding ja\_JP.utf8 locale and the target environment uses the encoding ja\_JP locale, some file names may not be handled correctly in the target.]]
- [[When you move the configuration of a component, the scripts replicate the topology of the source. For example, if the source domain contains Managed Servers server\_1 and server\_2 on Host A and Managed Servers server\_3 and server\_4 on Host B, you must specify a similar relationship between Managed Servers and hosts at the target. (You specify the hosts for each Managed Server in the move plan.)]]
- [[If a custom application uses an internal data source (for example, the application was created and deployed with an internal data source using JDeveloper), the internal data source is not migrated during the pasteConfig operation. ]
 

[To work around this, create an external data source in the domain, modify the application to use that data source, and deploy the application again.]]
- [[The JDK used in the source and target must be the same type. For example, if the source uses Java SE, the target must use Java SE. In addition, the vendor must be the same. For example, if the source uses an Oracle JDK, the target must use a JDK from Oracle. ]
  - [[The JDK used in the source and target must be the same type. For example, if the source uses Java SE, the target must use Java SE.]]
  - [[The vendor used in the source and target must be the same. For example, if the source uses an Oracle JDK, the target must use a JDK from Oracle. ]]
  - [[The major version of the JDK used in the source and target must be the same. For example, if the source uses version 1.7, the target must use 1.7.]]]]

- [[If there is not enough space in the temporary directory when you are moving an entity, an error is returned, noting the space needed. To work around this problem, specify a different location for the temporary directory by using the T2P\_JAVA\_OPTIONS environment variable as described in [Section A.1.1].]]
- [[If you have moved your environment and executed the pasteBinary script using a custom inventory location (using the invPtrLoc parameter), you must invoke runInstaller with the following argument:]
 

```
[[-invPtrLoc#[custom_inventory_pointer_location]]
[]]]
```
- [[When you are moving Oracle Platform Security Services and the data is moving from LDAP to LDAP, the source and target LDAP domain component hierarchy must be same. If it is not, the Oracle Platform Security Services data movement will fail. For example, if the source is hierarchy is configured as [dc=us, dc=com], the target LDAP must have the same domain component hierarchy.]]
- [[If your environment contains both IPV4 and IPV6 protocols, when you update the move plan, any IP addresses must use the IPV4 protocol.]]]]

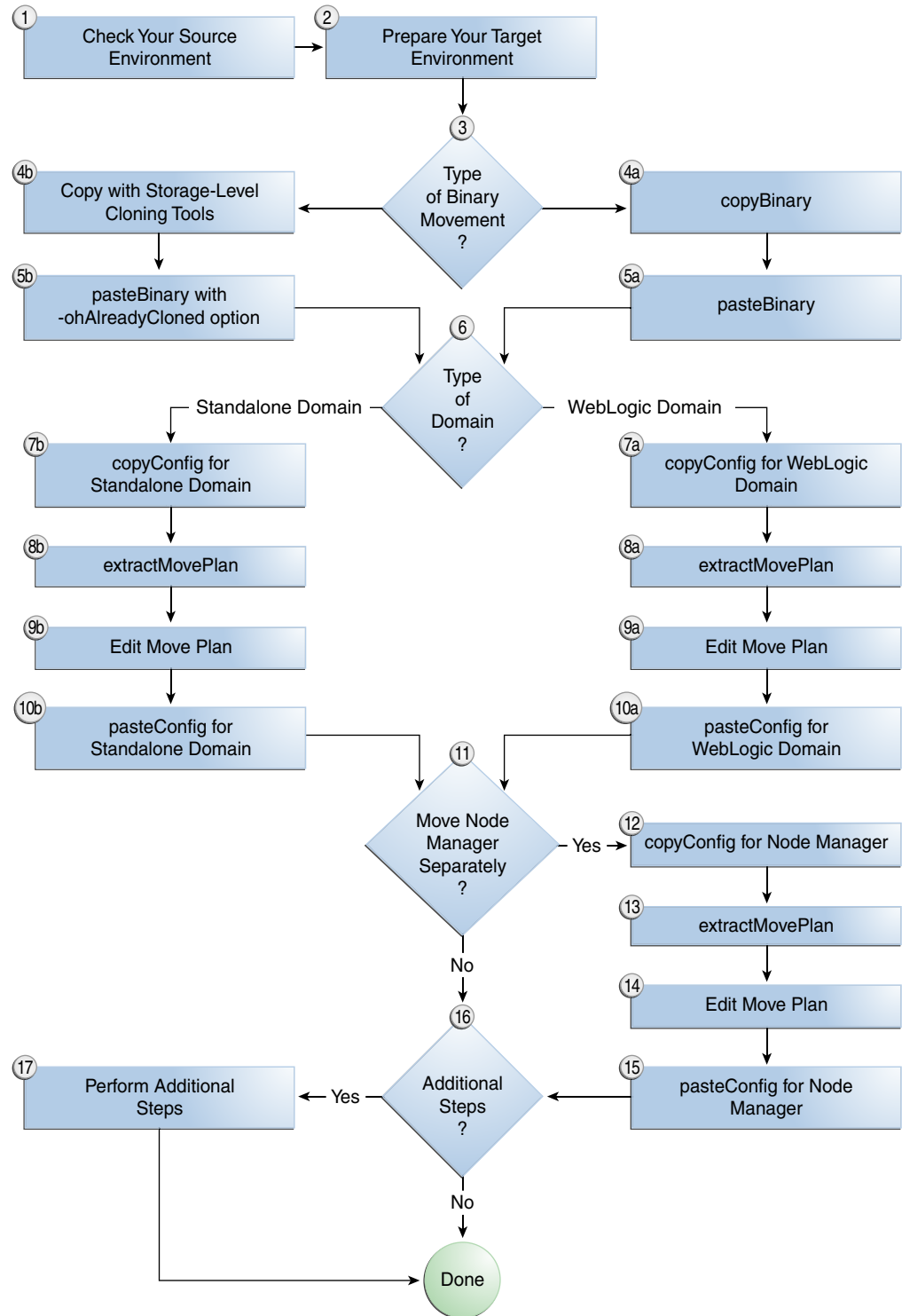
## 19.2.5 [[Overview of Procedures for Moving from a Source to a Target Environment]

[This section describes the general steps in moving installations from a source environment to a target environment. [Figure 19–1] shows a flowchart illustrating the steps.]

**Figure 19–1** *[[Flowchart for Moving Your Environment]*

[





]]

[The general steps are:]

1. [[[Check your source environment. See [Section 19.2.2](#).]]]
2. [[Prepare your target environment. See [Section 19.2.3](#).]]]
3. [[[If your environment uses a database, create a new database. See [Section 19.3.1](#).]]]

4. [[Move a copy of the binary files in an Oracle home from the source environment to the target environment ]
  - [[[[Using the copyBinary and pasteBinary scripts, as described in [\[Section 19.3.2\].](#)]]]]
  - [[Using storage-level cloning tools, if supported by your environment, to create a copy of an existing disk volume and move it to a different location. Then, you use the pasteBinary script to transform the target Oracle home to a proper Oracle home, creating or updating the necessary inventory information, file permissions, and string substitutions for the correct ORACLE\_HOME path. See [\[Section 19.3.3\].](#)]]

[You can use this method if your environment is located on one disk volume.]]]]
5. [[Move a copy of the configuration of the domain and components. In most cases, you use the copyConfig, extractMovePlan, and pasteConfig scripts. The procedure you follow differs depending on your topology:]
  - [[[[To move the configuration of a WebLogic Server domain containing only Java components, or Java components and system components, see [\[Section 19.3.4\].](#)]]]]
  - [[To move the configuration of a standalone domain containing system components, see [\[Section 19.3.5\].](#)]]]]
6. [[In certain situations as described in [\[Section 19.3.6\]](#), you must separately move a copy of the configuration of Node Manager if it is configured in the source environment.]]
7. [[Start the servers and components. See [\[Section 19.3.8\].](#)]]]]

## 19.3 [[Common Procedures for Moving to a Target Environment]

[Many of the Oracle Fusion Middleware components use some of the same procedures to move from a source environment to a target environment. Note, however that not all components use all or some these procedures. In addition, some components may require additional steps. You **[must]** check [\[Table 19–2\]](#) to see if there are additional steps you need to take when moving a particular component.]

[This section describes the common procedures and contains the following topics:]

- [[[[[Installing the Database on the Target Environment](#)]]]]
- [[[[[Moving the Oracle Home and the Binary Files Using the Scripts](#)]]]]
- [[[[[Moving the Configuration of a WebLogic Server Domain](#)]]]]
- [[[[[Moving the Configuration of a Standalone Domain](#)]]]]
- [[[[[Moving the Configuration of Node Manager](#)]]]]
- [[[[[Configuring Users and Groups](#)]]]]

[The procedures in this section assume that you are using the standard installation topology. This topology consists of a WebLogic Server domain that contains an Administration Server and a cluster of two Managed Servers on one host or a standalone domain containing system components]

[If you have distributed your topology across multiple machines, see [\[Section 19.6\].](#)]

---



---

**[Note]:** [In the scripts used in these procedures and in the move plans, you often need to provide files containing passwords. To generate a file that contains an obfuscated password, use the `obfuscatePassword` script, which is described in [\[Section A.1.1.10\]](#).]

---



---

### 19.3.1 [[Installing the Database on the Target Environment]

[Some components, such as Oracle Application Development Framework, may use a database to store metadata.]

[Note that the database in the target environment must be the same type of database as in the source environment. For example, if the database in the source environment is an Oracle Database, the database in the target environment must be an Oracle Database. The database on the target environment should be the same version as on the source environment.]

[To install a new database:]

1. [[[Install and configure the database software.]]]
2. [[Create the required schemas in the target database using RCU. See [\[\[Creating Schemas with the Repository Creation Utility\]](#).]]]
3. [[Create any custom schemas used by your applications. For example, if your application uses a custom schema in the source environment, create the schema in the target environment.]]]]

### 19.3.2 [[Moving the Oracle Home and the Binary Files Using the Scripts]

[You can move a copy of the Oracle home to the target environment using the `copyBinary` and `pasteBinary` scripts:]

- [[[The `copyBinary` script prepares the source and creates an archive. It also records the file permissions of the Oracle home.]

[The archive contains the Oracle home, including the product homes, such as Oracle WebLogic Server home and the Oracle HTTP Server home. ]]

- [[The `pasteBinary` script checks to see that the prerequisites are met at the destination. It extracts the files from the archive file, registers the Oracle home with the Oracle inventory.]

[The script then restores the file permissions and relinks any files if necessary. ]]

[Note the following:]

- [[[The `copyBinary` and `pasteBinary` scripts do not carry over all the dependencies of the source Oracle home and the product homes, such as the WebLogic Server home, such as loadable modules or application-specific libraries to the target home, because the scripts proceed by copying the Oracle home and the entire source product homes to the destination Oracle home. Any files outside the source WebLogic Server or Oracle home are not automatically copied. Hence, any applications that refer to files outside the source WebLogic Server or Oracle home may not work properly in the target home.]]]
- [[When you copy an Oracle home, only the read-only portions of the Oracle home are copied. Any user configuration files, such as the `user_projects` directory, are excluded from the archive. The WebLogic Server domain is not copied. (Use the `copyConfig` and `pasteConfig` scripts to copy the domain.))]]]
- [[You cannot move an Oracle Home if its path is a symbolic link.]]]

“ **[See]:** [\[Table A-1\]](#) for the location of the scripts used in this section.]

[To move the Oracle home:]

1. [\[\[\[At the source, execute the copyBinary script, which copies the Oracle home and the product homes, such as the WebLogic Server home, contained within the Oracle home. \]](#)

[See [\[Section A.1.1.1\]](#) for the syntax of the copyBinary script.]

[For example, to copy an Oracle home that is located at /scratch/oracle/Oracle\_home1, use the following command:]

```
[copyBinary.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/oh_copy.jar]
[#####-sourceOracleHomeLoc#/scratch/oracle/Oracle_home1]
[#####-invPtrLoc#/scratch/oracle/oraInst.loc]
[]]
```

2. [\[\[If you are copying the Oracle home to a different host, copy the archive file to that system, or if you are using storage-level cloning, copy the snapshot copy of the volume to the target system and mount the volume. \]\]](#)
3. [\[\[Copy the pasteBinary script and the cloningclient.jar file to the target system and ensure that they have execute permission. \]\]](#)

[The cloningclient.jar file is located in:]

```
[(UNIX)#[ORACLE_COMMON_HOME]/jlib/cloningclient.jar]
[(Windows)#[ORACLE_COMMON_HOME]\jlib\cloningclient.jar]
[]]
```

[Do **[not]** copy the other scripts, such as pasteConfig. Those scripts are generated when you extract the files, as in step [5].]

4. [\[\[On Linux and UNIX, if the target system does not contain any installed Oracle products, you must create an oraInst.loc file, specifying a group whose members are given access to write to the Oracle inventory \(oraInventory\), and where you want to put Oracle inventory. For example, the oraInst.loc file could contain the following:\]](#)

```
[[inst_group=dba]
[inventory_loc=/scratch/oracle1/oraInventory]
[]]
```

[Then, if the location is not the default location (/etc/oraInst.loc), use the -invPtrLoc option to the pasteBinary script to specify the location of the oraInst.loc file.]]

5. [\[\[At the target, extract the files from the archive using the pasteBinary script. See \[\\[Section A.1.1.2\\]\]\(#\) for the syntax of the pasteBinary script.\]](#)

“ **[Note]:** [If the *[parent]* directory for the Oracle home does not exist, the pasteBinary script will create it.]

[The actual directory for the Oracle home (for example, Oracle\_Home\_prod) cannot exist.]

[For example, to apply the archive to the directory /scratch/oracle/ORACLE\_HOME\_prod, use the following command:]

```
[pasteBinary.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
[#####-archiveLoc##/tmp/oh_copy.jar#]
[#####-targetOracleHomeLoc#/scratch/oracle/ORACLE_HOME_prod#]
```

- ```

#####-targetOracleHomeName#ORACLE_HOME_prod]
[]]
[The Oracle home is extracted to /scratch/oracle/ORACLE_HOME_prod and the
product homes are extracted under it with the same names as that of the source
product home names.]]

```
6. [[The copyBinary and pasteBinary scripts copy and paste any domain directories that are in the Oracle home, but not under the user_projects directory. These domain directories are not properly configured, so that they are not functional.]

[Delete the domain directories from the target before you run the pasteConfig command, which will recreate properly configured domain directories on the target environment, as described in [Section 19.3.4].]]
 7. [[At the target, if the Node Manager is per machine and is located under the Oracle home, delete the Node Manager directory and the files in it.]

[In this situation, you will move the Node Manager configuration in [Section 19.3.6].]]]

19.3.3 [[Moving the Oracle Home and Binary Files Using Storage-Level Cloning Tools]

[As an alternative to the copyBinary script, you can use storage-level cloning tools, such as Oracle Solaris ZFS or NetApp Flex Cloning, to create a copy of an existing disk volume and move it to a different location.]

[You can use this method if your environment is located on one disk volume.]

[To move the Oracle home and binary files using storage-level cloning tools:]

1. [[Use the cloning tool to replicate the disk volume to the target environment.]

[Refer to the documentation for your disk volume for more specific information.]]
2. [[At the target, use the pasteBinary script using the -ohAlreadyCloned option. With this option, the pasteBinary script creates or updates the necessary inventory information, file permissions, and string substitutions for the correct ORACLE_HOME path.]

[See [Section A.1.1.2] for the syntax of the pasteBinary script.]

[For example, to apply the archive to the directory /scratch/oracle/ORACLE_HOME_prod, use the following command:]

```

[ [pasteBinary.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
#####-ohAlreadyCloned##true#]
#####-targetOracleHomeLoc#/scratch/oracle/ORACLE_HOME_prod#]
#####-targetOracleHomeName#ORACLE_HOME_prod]
[]]]

```

3. [[At the target, if the Node Manager is "per machine" and is located under the Oracle home, delete the Node Manager directory and the files in it.]

[In this situation, you will move the Node Manager configuration in [Section 19.3.6].]]
4. [[Because the storage-level cloning tools copy the entire user directory, it copies not just the binary files, but also the domain directories if they have been configured in the source environment. However, these domain directories are not properly configured, so that they are not functional.]

[Delete the domain directories from the target before you run the pasteConfig command, which will recreate properly configured domain directories on the target environment, as described in [Section 19.3.4].]]]

19.3.4 [[Moving the Configuration of a WebLogic Server Domain]

[You can move a copy of the WebLogic Server domain configuration using the copyConfig, extractMovePlan, and pasteConfig scripts. This step moves a copy of the configuration, including the domain, the Administration Server and Managed Servers and any components in the domain. Then, it starts the Administration Server.]

[When you move the configuration of a component, the scripts replicate the topology of the source. For example, if the source domain contains Managed Servers server_1 and server_2 on Host A and Managed Servers server_3 and server_4 on Host B, you must specify a similar relationship between Managed Servers and hosts at the target. (You specify the hosts for each Managed Server in the move plan.)]

[The domain directory is local to each machine. The pasteConfig script is performed only on the Administration Server domain directory. Subsequently, if the Managed Servers are not in the same directory as the Administration Server, you must re-create the domain directory for those Managed Servers by using the Oracle WebLogic Server pack and unpack commands. For more information, see [[*Creating Templates and Domains Using the Pack and Unpack Commands*]].]

[Because, in most cases, the user-specific data is not the same in the target environment as in the source environment, this process does not move user-specific data.]

" **[See]:** [[Table A-1] for the location of the scripts used in this section.]

" **[Note]:** [By default, during the copyConfig and pasteConfig operations, the following, which specify the maximum heap size and the maximum permanent generation space, are set:]

```
[ [-Xmx512m#-XX:MaxPermSize=256m] ]
```

[You can modify the values using the T2P_JAVA_OPTIONS parameter of the copyConfig and pasteConfig scripts.]

[To move a copy of the domain configuration:]

1. [[[At the source, make sure that the Administration Server and all Managed Servers are started.]]]
2. [[At the source, make sure that the domain configuration is not set to automatically acquire locks. If you configured the domain in development mode, automatically acquiring locks is enabled. If you configured the domain in production mode, it is disabled by default. Take the following steps:]
 - a. [[[In the Administration Console, click **Preferences**.]]]
 - b. [[In the User Preferences tab, clear **Automatically Acquire Lock and Activate Changes**.]]]
 - c. [[Click **Save**.]]]
 - d. [[In the Change Center, click **Release Configuration**, if applicable.]]]
3. [[At the source, copy the domain configuration by executing the copyConfig script.]

[The copyConfig script is located in:]

```
[ ( (UNIX) # [ORACLE_COMMON_HOME] /bin/copyConfig.sh )
  ( (Windows) # [ORACLE_COMMON_HOME] \bin\copyConfig.cmd )
  [ ] ]
```

[See [Section A.1.1.3] for the syntax of the copyConfig script.]

[For example, to copy the configuration of the domain named WLS_domain1 in the Oracle home /scratch/oracle/Oracle_home1, use the following command:]

```
[copyConfig.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
#####-archiveLoc#/tmp/wls.jar]
#####-sourceDomainLoc#/scratch/oracle/domains/WLS_domain1]
#####-sourceOracleHomeLoc#/scratch/oracle/Oracle_home1]
#####-domainHostName#[example.com]]
#####-domainPortNum#8001]
#####-domainAdminUserName#[domain_admin_username]]
#####-domainAdminPasswordFile#/scratch/admin/passwd.txt]
#####-logDirLoc#/tmp/logs]
[]]
```

4. [[If you are copying the domain configuration to a different host, copy the archive file to that system.]]
5. [[At the source, extract the move plan from the archive, using the extractMovePlan script.]]

[See [Section A.1.1.6](#) for the syntax of the extractMovePlan script.]

[For example:]

```
[extractMovePlan.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
#####-archiveLoc#/tmp/wls.jar]
#####-planDirLoc#/tmp/Oracle/t2p_plans/wls]
[]]
```

[Note]: [You must extract a new move plan each time you use the copyConfig script even if no changes have been made to the source environment. The pasteConfig scripts checks that the move plan and archive match. If they do not, the script returns an error.]

6. [[Edit the move plan, modifying the properties to reflect the values for the target environment. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment. See [Table A-11](#) to find the list of properties for the type of component you are moving.]]
7. [[Copy the edited move plan to the target. (During the pasteConfig operation, you specify the location using the -movePlanLoc option.)]]
8. [[At the target, run the following script to generate obfuscated password files required by the move plan. Run the script for each password file.]

```
[ (UNIX)#[ORACLE_COMMON_HOME]/bin/obfuscatePassword.sh]
[ (Windows)#[ORACLE_COMMON_HOME]\bin\obfuscatePassword.cmd]
[]]
```

[The script prompts you to enter the password and the location where the password file is to be written.]]

9. [[At the target, if the -opssDataExport parameter was set to true during the copyConfig operation (it is set to true by default), you must set the following environment variable, which sets the initial and maximum heap size:]

```
[CONFIG_JVM_ARGS#" -Xmx2048M#-Xms2048M"]
[]]
```

10. [[At the target, extract the files from the archive using the pasteConfig script]]

[See [Section A.1.1.7](#) for the syntax of the script.]

[For example, to apply the archive to the Oracle home /scratch/oracle/Oracle_home1, use the following command:]

```

[[pasteConfig.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/wls.jar]
[#####-movePlanLoc#/tmp/Oracle/t2p_plans/wls/moveplan.xml]
[#####-targetDomainLoc#/scratch/oracle/config/domains/WLS_domain1]
[#####-targetOracleHomeLoc#/scratch/oracle/Oracle_home1/]
[#####-domainAdminPasswordFile#/scratch/pwd_dir/passwd.txt]
[#]]]

```

11. [[If Managed Servers are not located in the same directory as the Administration Server, you must re-create the domain directory for those Managed Servers by using the Oracle WebLogic Server pack and unpack commands. For more information, see [\[\[Creating Templates and Domains Using the Pack and Unpack Commands\]\]](#).]]

[When you complete this task, you may need to perform additional steps for some components, as described in [\[Section 19.4\]](#).]

19.3.5 [\[\[Moving the Configuration of a Standalone Domain\]](#)

[You can move the configuration of standalone domain containing system components. For example, you may have installed Oracle HTTP Server in a standalone domain.]

“ **[See]:** [\[\[Table A-1\]](#) for the location of the scripts used in this section.]

[To move the configuration of a standalone domain containing system components:]

1. [\[\[\[At the source, copy the configuration by executing the copyConfig script. \]](#)

[See [\[Section A.1.1.4\]](#) for the syntax of the copyConfig script.]

[For example, to copy the configuration of the domain named WLS_domain1 in the Oracle home /scratch/oracle/Oracle_home1, use the following command:]

```

[[copyConfig.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
[#####-archiveLoc#/tmp/a.jar]
[#####-sourceDomainLoc#/scratch/oracle/domains/WLS_domain1#]
[#####-sourceOracleHomeLoc#/scratch/oracle/Oracle_home1/]
[[]]]

```

2. [\[\[If you are copying the configuration to a different host, copy the archive file to that system. \]\]](#)
3. [\[\[At the source, extract the move plan from the archive created by the copyConfig script, using the extractMovePlan script. \]](#)

[See [\[Section A.1.1.6\]](#) for the syntax of the extractMovePlan script.]

[For example:]

```

[[extractMovePlan.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/wls.jar]
[#####-planDirLoc#/tmp/Oracle/t2p_plans/wls]
[[]]]

```

“ **[Note]:** [You must extract a new move plan each time you use the copyConfig script even if no changes have been made to the source environment. The pasteConfig scripts checks that the move plan and archive match. If they do not, the script returns an error.]

4. [[Edit the move plan, modifying the properties to reflect the values for the target environment. See [Table A-11] to find the list of properties for the type of component you are moving.]]
5. [[Copy the edited move plan to the target. (During the pasteConfig operation, you specify the location using the -movePlanLoc option.)]]
6. [[At the target, run the following script to generate obfuscated password files required by the move plan. Run the script for each password file.]

```
[[ (UNIX)#[ORACLE_COMMON_HOME] /bin/obfuscatePassword.sh
[(Windows)#[ORACLE_COMMON_HOME]\bin\obfuscatePassword.cmd
[]]
```

[The script prompts you to enter the password and the location where the password file is to be written.]]

7. [[At the target, extract the files from the archive using the pasteConfig script]

[See [Section A.1.1.7] for the syntax of the script.]

[For example, to apply the archive to the Oracle home /scratch/oracle/Oracle_home1, use the following command:]

```
[[pasteConfig.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/java_ee_cl.jar]
[#####-targetDomainLoc#/scratch/oracle/config/domains/dom_cl]
[#####-targetOracleHomeLoc#/scratch/oracle/Oracle_home1#]
[#####-movePlanLoc#/scratch/oracle/java_ee/move_plan.xml]
[#####-logDirLoc#/tmp/log]]]]]
```

19.3.6 [[Moving the Configuration of Node Manager]

[If Node Manager is configured in the source environment, you must separately move Node Manager in the following circumstances:]

- [[The Node Manager is "per machine".]]
- [[The Node Manager is "per domain" but its configuration is located outside of the domain directory.]]
- [[In an environment on multiple hosts, the Node Manager is "per domain" and its configuration is within the domain directory, but each host has customized Node Manager properties that are applicable to only that host.]]]

[If the Node Manager is per domain, the scripts for moving the domain also move the Node Manager.]

" [See]: [[Table A-1] for the location of the scripts used in this section.]

[To move the Node Manager configuration:]

1. [[At the source, copy the Node Manager configuration, by executing the copyConfig script.]

[See [Section A.1.1.5] for the syntax of the script. For example, use the following command:]

```
[[copyConfig.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/nm.jar]
[#####-sourceNMHomeLoc#/scratch/oracle/Oracle_
home1/wlserver/common/nodemanager]
[#####-logDirLoc#/tmp/logs]
[]]]]
```

2. [[If you are copying the Node Manager to a different host, copy the archive file to that system.]]
3. [[At the source, extract the move plan from the archive, using the extractMovePlan script.]]

[See [Section A.1.1.6](#) for the syntax of the extractMovePlan script.]

[For example:]

```
[[extractMovePlan.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/nm.jar]
[#####-planDirLoc#/tmp/Oracle/t2p_plans/nm]
[]]]
```

4. [[Edit the move plan, modifying the properties to reflect the values for the target environment. See [Table A-14](#) to find the list of properties for Node Manager.]]
5. [[Copy the edited move plan to the target. (During the pasteConfig operation, you specify the location using the -movePlanLoc option.)]]
6. [[At the target, run the following script to generate obfuscated password files required by the move plan. Run the script for each password file.]

```
[[ (UNIX) #[ORACLE_COMMON_HOME] /bin/obfuscatePassword.sh]
[ (Windows) #[ORACLE_COMMON_HOME] \bin\obfuscatePassword.cmd]
[]]]
```

[The script prompts you to enter the password and the location where the password file is to be written.]]

7. [[At the target, extract the files from the archive using the pasteConfig script.]]

[See [Section A.1.1.9](#) for the syntax of the script.]

[For example, use the following command:]

```
[[pasteConfig#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/nm.jar]
[#####-targetNMHomeLoc#/scratch/oracle/Oracle_
home1/wlserver/common/nodemanager]
[#####-targetOracleHomeLoc#/scratch/oracle/Oracle_home1]
[#####-movePlanLoc#/tmp/Oracle/t2p_plans/nm/moveplan.xml]]]]]
```

19.3.7 [[Configuring Users and Groups]

[You must configure security in the new target environment. The steps you take depends on the configuration of your environment and application.]

[The target environment LDAP identity store may not use the same users and groups as the source environment, or it may already be populated with users and groups. Take the following steps only if the LDAP store is an Oracle Internet Directory LDAP store and you need to move users, groups, and passwords from the source environment to the target environment:]

1. [[Export the users and groups from LDAP identity store on the source environment, using the ldapsearch command. This produces an ldif file that you later import into the LDAP identity store in the target environment. The ldapsearch command is located in the ORACLE_HOME/bin directory of the Identity Management components. For example:]

```
[[ [ORACLE_HOME] /bin/ldapsearch#-h#[test_oid_host]#-p#[test_oid_port]#]
[##-D#"cn=orcladmin"#-w#" [test_orcladmin_passwd] "#-b#"cn=Users,dc=us"]
[]]]
```

2. [[Import the ldif file that you exported from the source environment into the target environment, using the ldapaddmt command, as shown in the following example. ([ORACLE_HOME] is the Oracle home for Identity Management.)]

```
[ [ORACLE_HOME/bin/ldapaddmt#-h#[production_oid_host]
###-p#[production_oid_port]#-D#"cn=orcladmin"
###-w#" [production_orcladmin_passwd] "#-r#-f#[ldif_filename]]]]]]
```

19.3.8 [[Starting Managed Servers and Components]

[When the movement procedure completes, the Administration Server is started. However, the Managed Servers, Node Manager, and the components are stopped. Take the following steps:]

1. [[Start Node Manager, as described in [Section 4.2.2].]]
2. [[Start the Managed Servers, as described in [Section 4.2.3]]]
3. [[Start components, as described in [Section 4.3].]]]]]

19.4 [[Additional Steps for Certain Components]

[[Table 19–2] shows whether any additional steps are needed to complete the movement of particular components:]

Table 19–2 []Components Requiring Additional Steps for Movement to a Different Environment

[Component]	[Additional Procedures]
[Oracle Application Development Framework]	[None]
[Oracle Coherence]	[None]
[Oracle Data Integrator]	[See [Section 19.4.1]]
[Oracle HTTP Server]	[None]
[Oracle User Messaging Service]	[None]
[Oracle Web Services Manager]	[None]
[Oracle WebLogic Server]	[None]

19.4.1 [[Additional Steps for Moving Oracle Data Integrator]

[Note the following additional steps that you must take when moving Oracle Data Integrator:]

- [[Create the required master and work repositories schemas in the target database using RCU. See the [[Creating Schemas with the Repository Creation Utility.]]]

[Make sure that the target work repository is created with the same type as the source repository (For example, if the source work repository is created as a development repository, the target work repository must also be created as a development repository).]

- [[When you run the copyConfig script, note the following:]
 - [[You must pass a configuration file to the copyConfig script. You pass this using the -additionalParams option with the argument odiCustomArg. For example:]

```
[ [./copyConfig.sh#-javaHome#/private/Middleware/jrockit_160_26_D1.2.0-5]
###-archiveLocation#/tmp/ar.jar#]
```

```

[###-sourceOracleHomeLoc#/private/Middleware#]
[###-sourceDomainLoc#/scratch/oracle/domains/base_domain#]
[###-domainHostName#host1.example.com#]
[###-domainPortNo#7001#]
[###-domainAdminUserName#weblogic#]
[###-domainAdminPasswordFile#/tmp/wls_pswd.txt#]
[###-additionalParams#odiCustomArg=/private/t2p/odiCustomArg.xml]
[]

```

[The file `odiCustomArg.xml` is the configuration file. A sample file is located in:]

```

[[ [ORACLE_HOME] / [ODI_Oracle_Home] / odi / plugin / t2p / odiCustomArg.xml ]
[]

```

[The configuration file that you pass to the script contains the connection information for all Oracle Data Integrator master repositories. The following shows a sample configuration file:]

```

[ [<?xml#version="1.0"#encoding="UTF-8"##?> ]
[ <config> ]
[ ###<masterRepositories> ]
[ #####<masterRepository> ]
[ #####<driver>oracle.jdbc.OracleDriver</driver> ]
[ #####<url>jdbc:oracle:thin:@localhost:1521:[sid_or_service_ ]
[ name] / example.com</url> ]
[ #####<schema>odi_master_11g</schema> ]
[ #####<schema_password_file>/tmp/all_pswd.txt</schema_password_file> ]
[ #####<supervisor>SUPERVISOR</supervisor> ]
[ #####<supervisor_password_file>/tmp/sup_pswd.txt</supervisor_ ]
[ password_file> ]
[ #####</masterRepository> ]
[ #####<masterRepository> ]
[ #####. . . . content#for#2nd#master#repository ]
[ #####</masterRepository> ]
[ #####</masterRepositories> ]
[ </config># ]
[]

```

[The following explains the entries in the configuration file:]

- * [[masterRepositories: Contains the list of ODI Master Repositories.]]
- * [[masterRepository: The section for ODI Master Repositories.]]
- * [[driver: The JDBC Driver to connect to the ODI Master Repository.]]
- * [[url: The JDBC URL to connect to the ODI Master Repository.]]
- * [[schema: The schema name for the ODI Master Repository.]]
- * [[schema_password_file: The path for the file containing the password for the schema.]]
- * [[supervisor: The supervisor user for the ODI Master Repository.]]
- * [[supervisor_password_file: The path for the file containing the password for the supervisor user.]]]]]
- [[The movement scripts update the physical architecture in the target environment according to the information you specified in the move plan. Review the following items in the physical architecture in the target environment before proceeding:]
 - [[Physical Agents: Change the host, port, and Web application context (for Java EE Agent) to match the configuration of the target environment.]]

- [\[\[Data Servers: Change the data server connection information \(JDBC, JNDI, data source name\) to match the configuration of the target environment.\]\]](#)
- [\[\[Physical Schemas: The schemas \(including file folder location\) defined for the data servers must match the configuration of the target environment.\]\]\]\]](#)
- [\[\[After you complete the movement, restart the Java EE agents in the target environment. These agents start processing the scheduled scenarios.\]\]\]\]](#)

19.5 [\[\[Incrementally Moving Artifacts\]](#)

[The movement scripts are intended for moving to a new target environment. They do not support moving artifacts to an already existing environment.]

[If you have already moved your environment to a new target, at some later time, you may want to move artifacts that have changed in your source environment to your target environment. For information about moving artifacts that have changed, see the documentation for the particular component, such as Oracle HTTP Server.]

19.6 [\[\[Moving Distributed Topologies\]](#)

[The following topics describe considerations when you have a distributed topology:]

- [\[\[\[Considerations with a Multiple Host Environment\]\(#\)\]\]\]\]](#)
- [\[\[\[Considerations in Moving to and from an Oracle RAC Environment\]\(#\)\]\]\]\]](#)

19.6.1 [\[\[Considerations with a Multiple Host Environment\]](#)

[If your domain is distributed across multiple hosts, you must take additional steps to complete the movement.]

[When you move the configuration of a component, the scripts replicate the topology of the source. For example, if the source domain contains Managed Servers server_1 and server_2 on Host A and Managed Servers server_3 and server_4 on Host B, you must specify a similar relationship between Managed Servers and hosts at the target. (You specify the hosts for each Managed Server in the move plan.)]

[These steps assume that you have taken the steps in [\[Section 19.3\]](#) on the Administration Server host:]

1. [\[\[If you do not use shared disks, use the pasteBinary command to create an Oracle home on the remote host, for example, Host B. You use the same archive that you created in \[\\[Section 19.3.2\\]\]\(#\) Step \[1\].\]\]](#)

[For example:]

```
[[pasteBinary.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
#####-archiveLoc##/tmp/oh_copy.jar#]
#####-targetOracleHomeLoc#/scratch/oracle/ORACLE_HOME_prod#]
#####-targetOracleHomeName#ORACLE_HOME_prod]
]]]
```

2. [\[\[Re-create the domain directory for the remote Managed Servers by using the Oracle WebLogic Server pack and unpack commands. For more information, see \[\\[\\[*Creating Templates and Domains Using the Pack and Unpack Commands*\\]\\]\]\(#\).\]\]\]\]](#)

19.6.2 [\[\[Considerations in Moving to and from an Oracle RAC Environment\]](#)

[If you are moving your environment to or from an Oracle Real Application Cluster (Oracle RAC) environment, note the following:]

- [[If you are moving from a source environment that is not an Oracle RAC environment to a target environment that uses Oracle RAC, the move plan will have one entry for a generic data source (for example mds-adf.) You update the move plan to point to one of the Oracle RAC instances and complete the move from the source environment to the target environment.]
[Then, you configure your target environment for Oracle RAC, as described in the *[High Availability Guide]*, especially ["Database Considerations."]]]
- [[Multi data sources are moved to the target environment, even though they are not listed in the move plan.]]
- [[If you are moving from a source environment that uses Oracle RAC to a target environment that does not use Oracle RAC, the move plan will have multiple entries for generic data sources. For example, if you have four Oracle RAC instances, you will have four generic data sources that are named mds-adf-rac1 through mds-adf-rac4. You update the move plan to point all generic data sources to the single non-RAC instance in the target environment.]]
- [[If you are moving from a source environment that uses Oracle RAC to a target environment that uses Oracle RAC, but you have more Oracle RAC instances in the target environment, the move plan will have multiple entries for generic data sources. For example, if you have three Oracle RAC instances on the source environment, you will have three generic data sources that are named mds-adf-rac1 through mds-adf-rac3. You have four Oracle RAC instances in the target environment. You update the move plan to point the generic data sources to the first three generic data sources in the target environment.]]
- [[If you are moving from a source environment that uses Oracle RAC to a target environment that uses Oracle RAC, but you have fewer Oracle RAC instances in the target environment, the move plan will have multiple entries for generic data sources. For example, if you have four Oracle RAC instances on the source environment, you will have four generic data sources that are named mds-adf-rac1 through mds-adf-rac4. You have three Oracle RAC instances in the target environment. You update the move plan to point the first three generic data sources to the three generic data sources in the target environment. You point the last generic data source to the third generic data source. (The third Oracle RAC instance will contain both mds-adf-rac3 and mds-adf-rac4).]
[Then, you can add an additional data source, as described in [\[Section 9.2.2.1\]](#).]]]

19.7 [[Recovering from Test to Production Errors]

[When you execute the pasteBinary or pasteConfig scripts and enter incorrect information in the move plan, the scripts return an error. In some cases, the scripts may have partially completed the paste operation. To recover, take the following actions, depending on the script that returned the error:]

- [[On Windows if you are using the Sun JDK, the copyBinary, pasteBinary, copyConfig, or pasteConfig operations may fail with the following error:]

```
[[java.nio.channels.OverlappingFileLockException]
[]]
```

[In this case, use the T2P_JAVA_OPTIONS to set the system property sun.nio.ch.disableSystemWideOverlappingFileLockCheck as shown in the following example:]

```
[[set#T2P_JAVA_OPTIONS=]
[-Dsun.nio.ch.disableSystemWideOverlappingFileLockCheck=true]
[]]
```

[Then, retry the operation.]

- [[If the pasteBinary script returns an error while moving the Oracle home directory at the target:]
 1. [[[Delete the target Oracle home.]]]
 2. [[Remove the Oracle home entry from the Oracle inventory, if it is present.]]
 3. [[For Windows, remove the shortcut for the Oracle home.]]]
- [[The copyConfig script requires that all servers be running, but that they are idle, so that no directories are being modified. If a server is not idle, the copyConfig script reports that the cloning operation completed successfully and the copyConfig error log file will remain at 0 bytes. However, the copyConfig standard log file will contain an error regarding writing to the packed_domain.jar. That error will cause the pasteConfig process to fail.]

[To work around this issue, wait for a short period of time, then retry the copyConfig operation again.]

- [[If the pasteConfig script returns an error while moving Java components:]
 1. [[[Stop all processes related to the domain.]]]
 2. [[Delete the following directories:]]


```
[[ [ORACLE_HOME]/user_projects/domains/[domain_name]]
[[ [ORACLE_HOME]/user_projects/applications/[domain_name]]
[]]]
```
 3. [[Drop the schemas and re-create them using RCU.]]]

[In addition, if the Oracle Platform Security reassociation failed:]

- [[[If you are moving from a file-based store to an LDAP store, specify a different value in the move plan.]]]
- [[For an LDAP store, delete the domain node.]]]
- [[For a database-based store, drop the schema and re-create it using RCU.]]]
- [[If you encounter an out-of-memory error when you are using the pasteConfig script, you can work around this in one of the following ways:]
 - [[[Increase the JVM heap size: Use the option -Xmx for maximum heap size, and -Xms for initial heap size. For example:]]]


```
[[ CONFIG_JVM_ARGS="-Xms512m#-Xmx1024m" ]
[]]]
```
 - [[Often, the Oracle WebLogic Server domain directory structure contains some large, unnecessary files, such as large older log files. You can delete these files, then run the copyConfig and pasteConfig scripts again.]]]
- [[If you are moving Oracle Identity Manager, the pasteConfig script may generate the following error:]

```
[ [NOTIFICATION:#PManager#instance#is#created#without#multitenancy#support#as]
[ JVM#flag#"oracle.multitenant.enabled"#is#not#set#to#enable#multitenancy]
[ support.]
[ Sep#24,#2013#10:26:55#PM]
[ oracle.security.jps.internal.config.xml.XmlConfigurationFactory]
[ initDefaultConfiguration]
[ SEVERE:#java.io.FileNotFoundException:#./config/jps-config.xml#(No#such#file]
[ or#directory)]
[ Sep#24,#2013#10:26:55#PM#oracle.mds]
[ NOTIFICATION:#Auditing#is#disabled#for#component#MDS.]
```

```
[Sep#24,#2013#10:26:55#PM#oracle.mds]
[NOTIFICATION:#PManager#instance#is#created#without#multitenancy#support#as]
[JVM#flag#"oracle.multitenant.enabled"#is#not#set#to#enable#multitenancy]
[support.]
[Sep#24,#2013#10:26:55#PM]
[oracle.security.jps.internal.config.xml.XmlConfigurationFactory]
[initDefaultConfiguration]
[SEVERE:#java.io.FileNotFoundException:./config/jps-config.xml#(No#such#file]
[or#directory)]
[Sep#24,#2013#10:26:55#PM#oracle.mds]
[NOTIFICATION:#Auditing#is#disabled#for#component#MDS.]
[Sep#24,#2013#10:26:55#PM#oracle.mds]
[#] ]
[This error is benign and can be safely ignored.]]]]]
```


Part IX

Appendixes

This part contains the following appendixes:

- [Appendix A, "Movement Scripts and Move Plans"](#)
- [Appendix B, "Oracle Fusion Middleware Command-Line Tools"](#)
- [Appendix C, "URLs for Components"](#)
- [Appendix D, "Port Numbers"](#)
- [Appendix E, "Using Oracle Fusion Middleware Accessibility Options"](#)
- [Appendix F, "Viewing Release Numbers"](#)
- [Appendix G, "orapki"](#)
- [Appendix H, "Troubleshooting Oracle Fusion Middleware"](#)

[[Movement Scripts and Move Plans]]

[Oracle Fusion Middleware provides a series of scripts that you can use to move your environment, for example replicating a test environment to a production environment. The scripts enable you to copy an Oracle home and Oracle WebLogic Server domains, as well as the configuration of certain Oracle Fusion Middleware components, such as Oracle HTTP Server. This appendix explains the scripts you can use to move these entities. It also describes the move plan properties that you edit when you move your environment.]

[This appendix contains the following sections:]

- [[[[Section A.1, "Understanding the Movement Scripts"]]]]
- [[[[Section A.2, "Modifying Move Plans"]]]]

A.1 [[Understanding the Movement Scripts]]

[The movement scripts copy the binary files of an Oracle home and the configuration of a domain and its components from a source environment and paste them at the target environment.]

[Use these scripts in conjunction with the procedures described in [Chapter 19].]

[Oracle Fusion Middleware uses the following jar file to execute the scripts necessary to move the binary and configuration files:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/jlib/cloningclient.jar]
[(Windows)#[ORACLE_COMMON_HOME]\jlib\cloningclient.jar]
[]]
```

[[Table A-1] shows the scripts you use to move an Oracle home or a domain and its components.]

Table A-1 []Movement Scripts

[TO:]	[Script]	[See:]
[Copy the binary files of the source Oracle home]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/copyBinary.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\copyBinary.cmd]]	[[Section A.1.1.1]]
[Apply the copied Oracle home to the target]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/pasteBinary.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\pasteBinary.cmd]]	[[Section A.1.1.2]]
[Copy a WebLogic Server domain and component configuration]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/copyConfig.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\copyConfig.cmd]]	[[Section A.1.1.3]]
[Copy a standalone domain and component configuration]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/copyConfig.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\copyConfig.cmd]]	[[Section A.1.1.4]]

Table A-1 *[(Cont.)]Movement Scripts*

[TO:]	[Script]	[See:]
[Copy the Node Manager configuration]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/copyConfig.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\copyConfig.cmd]]	[[Section A.1.1.5]]
[Extract a move plan from the domain or component]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/extractMovePlan.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\extractMovePlan.cmd]]	[[Section A.1.1.6]]
[Apply the copied configuration for the WebLogic Server domain and component configuration to the target]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/pasteConfig.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\pasteConfig.cmd]]	[[Section A.1.1.7]]
[Apply the copied standalone domain and component configuration to the target]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/pasteConfig.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\pasteConfig.cmd]]	[[Section A.1.1.8]]
[Apply the copied configuration for the Node Manager to the target]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/pasteConfig.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\pasteConfig.cmd]]	[[Section A.1.1.9]]
[Generate a file containing an obfuscated password]	[[(UNIX)#[ORACLE_COMMON_HOME]/bin/obfuscatePassword.sh] [(Windows)#[ORACLE_COMMON_HOME]\bin\obfuscatePassword.cmd]]	[[Section A.1.1.10]]

[To view the help on any of these scripts, use the -help option. For example:]

```
[ [./pasteConfig.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#-help]
[ ]]
```

[Note that the help shows the UNIX version of the parameter values. For other platforms, such as Windows, change the parameter values for the platform.]

"

[Note]: [During the copyConfig and pasteConfig operations, the following parameters, which specific the maximum heap size and the maximum permanent generation space, are set:]

```
[ [-Xmx512m]
[ -XX:MaxPermSize=256m]
[ ]]
```

[You can override these values using the T2P_JAVA_OPTIONS argument, as described in the following text.]

[To specify additional Java options, define the T2P_JAVA_OPTIONS environment variable and specify the options in the variable definition. The following examples set the value for the Java temp directory:]

- **[[[On Linux or UNIX:]**

```
[ [setenv#T2P_JAVA_OPTIONS#" -Djava.io.tmpdir=/home/t2p/temp" ]
[ export#T2P_JAVA_OPTIONS ]
[ ] ]]
```

- **[[[On Windows:]**

```
[ [set#T2P_JAVA_OPTIONS#" -Djava.io.tmpdir=c:\home\t2p\temp" ] ] ]]
```

[Note]: [A Universal Uniform Naming Convention (UNC) path is not supported on Windows. For example, the following is not supported:]

```
[[\\[host_name]\oracle\java\win64\jdk6\jre\bin\java]]
```

A.1.1 [[Movement Scripts Syntax]

[The following topics describe the syntax of the movement scripts. The options are described in the tables that follow the syntax.]

- [[[[copyBinary Script]]]
- [[pasteBinary Script]]]
- [[copyConfig Script for Oracle WebLogic Server Domains]]]
- [[copyConfig Script for Standalone Domains]]]
- [[copyConfig Script for Node Manager]]]
- [[extractMovePlan Script]]]
- [[pasteConfig Script for Oracle WebLogic Server Domains]]]
- [[pasteConfig Script for Standalone Domains]]]
- [[pasteConfig Script for Node Manager]]]
- [[obfuscatePassword Script]]]

[Note]: []

- [[All movement scripts ask if you want to continue whenever you specify the [-silent#false] option. To continue, you must type [yes], which is not case sensitive. Any words other than [yes] causes the script to abort. Also note that, in silent mode, the scripts generate an error if you do not provide passwords where they are needed.]]
- [[Most options have shortcut names, as described in the tables later in this chapter.]]
- [[The value of options must not contain a space. For example, on Windows, you cannot pass the following as a value to the -archiveLoc option:]

```
[[C:\tmp\Archive#Files]
[]]
```

[However, the value of the JavaHome option can contain a space.]]

- [[On Windows, unless the command prompt uses MKS or another application to support Unix commands, and if the values contain a Windows-specific delimiter, such as an equals sign (=) or comma (,) you must wrap the entire value in double quotation marks ("). For example:]

```
[[-additionalParams#"search.encrypt.key=C:\T2P\encrypt.txt"#]
[]]]
```

- [[The vendor and version of Java used in the javaHome option must match the vendor and version of the JAVA_HOME property defined in the following file (note the period (.) before the filename):]

```
[[ORACLE_HOME\wlserver/.product.properties]]]
```

A.1.1.1 [[copyBinary Script]

[Creates an archive file of the source Oracle home, by copying the binary files of that Oracle home, including its WebLogic Server home, into the archive file.]

[The copyBinary script is located in:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/copyBinary.sh]
[(Windows)#[ORACLE_COMMON_HOME]\bin\copyBinary.cmd]
[]]
```

[The syntax is:]

```
[[copyBinary#-javaHome#[path_of_jdk]
#####-archiveLoc#[archive_location]#]
#####-sourceOracleHomeLoc#[ORACLE_HOME_location]]
#####[-logDirLoc#[log_dir_path]]]
#####[-silent#{true#|#false}]]]
#####[-ignoreDiskWarning#{true#|#false}]]]
[]]
```

[The following example shows how to create an archive of an Oracle home on Linux:]

```
[[copyBinary.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
#####-archiveLoc#/tmp/oh_copy.jar]
#####-sourceOracleHomeLoc#/scratch/oracle/Oracle_home1#]]]
```

[Note]: [When you execute the script, you must specify a matching Java home. That is, if the Oracle homes are 64 bit, you must specify a 64-bit Java home. If the Oracle homes are 32 bit, you must specify a 32-bit Java home.]

[[Table A-2] describes the options for the copyBinary script.]

Table A-2 []Options for the copyBinary Script

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-javaHome]	[NA]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[-archiveLoc]	[-al]	[The absolute path of the archive location. Use this option to specify the location of the archive file to be created with the copyBinary script.] [The archive location must not exist.] [Ensure that the archive location is not within the Oracle home structure.]	[Mandatory]
[-sourceOracleHomeLoc]	[-soh]	[The absolute path of the Oracle home to be archived. You can only specify one Oracle home.] [In previous releases, this option was sourceMWHomeLoc. That option is deprecated.]	[Mandatory]
[-invPtrLoc]	[-ipl]	[This option is deprecated. If you specify it, it will be ignored. On UNIX and Linux, the absolute path to the Oracle Inventory pointer.]	[Ignored]
[-logDirLoc]	[-ldl]	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
[-silent]	[NA]	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of [false]. To continue, you must type [yes], which is not case sensitive. Typing anything other than yes causes the script to abort.]	[Optional]
[-ignoreDiskWarning]	[-idw]	[Specifies whether the operation ignores a warning that there is not enough free space available. The default is false.] [You may need to use this flag if the target is NFS mounted or is on a different file system, such as Data ONTAP.]	[Optional]

A.1.1.2 [[pasteBinary Script]

[Applies the archive to the target destination, by pasting the binary files of the source Oracle home to the target environment. You can apply the archive to the same host or a different host.]

[Before you execute this script, see [Section 19.2.3] for information about steps you may need to take to prepare your target environment.]

[The pasteBinary script is located in:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/pasteBinary.sh]
```

```
[ (Windows)#[ORACLE_COMMON_HOME]\bin\pasteBinary.cmd]
[]]
```

[The syntax is:]

```
[ [pasteBinary#-javaHome#[path_of_jdk]]
[#####-archiveLoc#[archive_location]]
[#####-targetOracleHomeLoc#[target_Oracle_Home_location]]
[#####-targetOracleHomeName#[Oracle_home_name]]
[#####[-ouiParam#key1=value], #key2=value]]
[#####[-ohAlreadyCloned#{true#|false}]]
[#####[-executeSysPrereqs#{true#|false}]]
[#####[-invPtrLoc#[Oracle_InventoryLocation]]]
[#####[-logDirLoc#[log_dir_path]]]
[#####[-silent#{true#|false}]]
[#####[-ignoreDiskWarning#{true#|false}]]
[]]
```

[The following example shows how to apply the archive to the directory /scratch/oracle/Oracle_home_prod, on Linux:]

```
[ [pasteBinary.sh#-javaHome#[scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc##/tmp/oh_copy.jar#]
[#####-targetOracleHomeLoc#[scratch/oracle/Oracle_home_prod#]
[#####-targetOracleHomeName#Oracle_home_prod#]
[]]
```

[[Table A-3] describes the options for the pasteBinary script.]

Table A-3 [Options for the pasteBinary Script

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-javaHome]	[NA]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[-archiveLoc]	[-al]	[The absolute path of the archive location. Use this option to specify the location of the archive file created with the copyBinary script.] [The location must exist.] [This option is mutually exclusive with the -ohAlreadyCloned option.]	[Mandatory]
[-targetOracleHomeLoc]	[-toh]	[The absolute path of the target Oracle home. It will be used to configure the domain.] [Ensure that the Oracle home directory does not exist at that location, or if it does, it is an empty directory. Otherwise, the script returns an error message.] [The -targetOracleHomeLoc cannot be inside another Oracle home.]	[Mandatory]
[-targetOracleHomeName]	[-tohn]	[The name for the Oracle home. This name is used to register the Oracle home with Oracle Inventory,]	[Optional]
[-ouiParam]	[-op]	[Additional variables to be passed to Oracle Universal Installer, which is run as part of this script. You must pass the variables as key=value pairs. Separate multiple variables with commas.] [On Windows, surround the entire value pair with double quotation marks (").]	[Optional]

Table A-3 [(Cont.)]Options for the *pasteBinary* Script

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-ohAlreadyCloned]	[-ohac]	[A flag specifying that the script reconfigure an already existing Oracle home that was created using a storage-level cloning tool. If this flag is set to true, then the target Oracle home should exist and it should contain Oracle home binaries.] [Valid values are [true] and [false]. The default is [false].] [You cannot use this option when you use the -archiveLoc option.]	[Optional]
[-executeSysPrereqs]	[-esp]	[Specifies whether the pasteBinary operation checks the prerequisites of the Oracle home. The default is that it checks the prerequisites. To specify that it does not check the prerequisites, specify this option with the value [false].]	[Optional]
[-invPtrLoc]	[-ipl]	[On UNIX and Linux, the absolute path to the Oracle Inventory pointer. Use this option if the inventory location is not in the default location, so that the operation can register the Oracle homes with the central Oracle inventory specified in the Oracle Inventory pointer file.] [If the oraInst.loc is not present at default location, you must create this file either at default location as a root user or at any other location as a root or normal user. The following shows an example of the contents of the file:] [[inventory_loc=/scratch/oraInventory]] [If the directory specified as the inventory_loc does not exist, the operation will create it.] [You must have write permission to the inventory location.] [On AIX and Linux, the default location is /etc/oraInst.loc. In other UNIX platforms, the default location is /var/opt/oracle/oraInst.loc] [This parameter is only supported on UNIX. On Windows, if you specify this parameter, the script returns an error.]	[Optional, if the inventory is in the default location. Otherwise, it is mandatory on Linux.]
[-logDirLoc]	[-ldl]	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
[-silent]	[NA]	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of [false]. To continue, you must type [yes], which is not case sensitive. Typing anything other than yes causes the script to abort.]	[Optional]
[-ignoreDiskWarning]	[-idw]	[Specifies whether the operation ignores a warning that there is not enough free space available. The default is false.] [You may need to use this flag if the target is NFS mounted or is on a different file system, such as Data ONTAP.]	[Optional]

A.1.1.3 [[copyConfig Script for Oracle WebLogic Server Domains]

[Creates a configuration archive that contains the snapshot of the configuration of a WebLogic Server domain. The underlying components of a WebLogic Server domain

retain their configuration information in different data stores, such as a file system, Oracle Metadata Services (MDS), LDAP, or a database.]

[You must run the copyConfig script for each WebLogic Server domain in the source environment. A configuration archive is created for each source domain.]

[The Administration Server and all Managed Servers in the domain must be started when you run the script.]

[The copyConfig script is located in:]

```
[ (UNIX)#[ORACLE_COMMON_HOME]/bin/copyConfig.sh]
[(Windows)#[ORACLE_COMMON_HOME]\bin\copyConfig.cmd]
[]]
```

[The syntax is:]

```
[copyConfig##-javaHome#[path_of_jdk]]
[#####-archiveLoc#[archive_location]#]
[#####-sourceDomainLoc#[domain_location]]
[#####-sourceOracleHomeLoc#[Oracle_home_location]]
[#####-domainHostName#[domain_host_name]]
[#####-domainPortNum##[domain_port_number]#]
[#####-domainAdminUserName#[domain_admin_username]]
[#####-domainAdminPasswordFile#[domain_admin_password_file]]
[#####[-mdsDataExport#{true#|#false}]]
[#####[-opssDataExport#{true#|#false}]#]
[#####[-trustKeyStoreLoc#[custom_trust_keystore_path]]]
[#####[-additionalParams#[property1]=[value1][[, property2]=[value2]]]]
[#####[-logDirLoc#[log_dir_path]]]
[#####[-silent#{true#|#false}]]
[]]
```

[The following example copies the configuration of a WebLogic Server domain:]

```
[copyConfig.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
[#####-archiveLoc#/tmp/a.jar]
[#####-sourceDomainLoc#/scratch/oracle/config/domains/WLS_domain]
[#####-sourceOracleHomeLoc#/scratch/oracle/Oracle_home1]
[#####-domainHostName#myhost.example.com#]
[#####-domainPortNum#7001]
[#####-domainAdminUserName#weblogic#]
[#####-domainAdminPasswordFile#/home/oracle/password_file.txt#]
[]]
```

[[Table A-4] describes the options for the copyConfig script for Oracle WebLogic Server domains.]

Table A-4 []Options for the copyConfig Script for a Oracle WebLogic Server Domains

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-javaHome]	[NA]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[-archiveLoc]	[-al]	[The absolute path of the archive location. Use this option to specify the location of the archive file to be created by the copyConfig script.]	[Mandatory]
[-sourceDomainLoc]	[-sd]	[The absolute path of the source domain containing the Java component.] [Note that on Windows, you should not include a backslash at the end of the path.]	[Mandatory]
[-sourceOracleHomeLoc]	[-soh]	[The absolute path of the source Oracle home.]	[Mandatory]

Table A-4 *[(Cont.)]Options for the copyConfig Script for a Oracle WebLogic Server Domains*

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-domainHostName]	[-dhn]	[The name of the host on which the domain is configured.]	[Mandatory]
[-domainPortNum]	[-dpn]	[The port number of the Administration Server for the domain.]	[Mandatory]
[-domainAdminUserName]	[-dau]	[The name of the administrative user for the domain.]	[Mandatory]
[-domainAdminPassword File]	[-dap]	[The absolute path of a secure file containing the password for the administrative user for the domain on the source environment. You must provide a password file, even if you are not changing the configuration.]	[Mandatory]
[-mdsDataExport]	[-mde]	[Specifies whether to export the application MDS metadata to the archive so that it can be imported into the target. The default is true.] [Specify false if you do not want to export the application MDS metadata.] [If this option is set to true, the subsequent pasteConfig script that copies the component to the target imports the application MDS metadata to the target.]	[Optional]
[-opssDataExport]	[-ode]	[Specifies whether to export the Oracle Platform Security Services data. The default is true.] [If this option is set to true, the subsequent pasteConfig script that copies the component to the target imports the Oracle Platform Security Services data to the target.] [[Note:] If this option is set to true, you must set the following environment variable before you run the pasteConfig script:] [[CONFIG_JVM_ARGS# " -Xmx2048M#-Xms2048M"]]	[Optional]
[-trustKeyStoreLoc]	[-tkl]	[The absolute path of the trust keystore location. Use this parameter if the domainPortNum is an SSL port and the server is configured with custom identity and custom trust keystores.]	[]
[-logDirLoc]	[-ldl]	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
[-additionalParams]	[-ap]	[An additional parameter and its value to be passed to the script. You must pass the variables as key=value pairs. Separate multiple variables with commas.] [On Windows, surround the entire value pair with double quotation marks (").]	[Optional]
[-silent]	[NA]	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of [false]. To continue, you must type [yes], which is not case sensitive. Typing anything other than yes causes the script to abort.]	[Optional]

A.1.1.4 [[copyConfig Script for Standalone Domains]

[Creates a configuration archive that contains the snapshot of the configuration of a standalone domain. The underlying components of the domain retain their configuration information in different data stores, such as a file system or a database.]

[You must run the copyConfig script for each domain in the source environment. A configuration archive is created for each source domain.]

[The copyConfig script is located in:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/copyConfig.sh]
[(Windows)#[ORACLE_COMMON_HOME]\bin\copyConfig.cmd]
[]]
```

[The syntax is:]

```
[[copyConfig##-javaHome#[path_of_jdk]
#####-archiveLoc#[archive_location]#]
#####-sourceDomainLoc#[domain_location]]
#####-sourceOracleHomeLoc#[Oracle_home_location]]
#####[-additionalParams#[property1]=[value1][[, property2]=[value2]]]
#####[-logDirLoc#[log_dir_path]]]
#####[-silent#{true#|#false}]]
[]]
```

[The following example copies the configuration of a standalone domain:]

```
[[copyConfig.sh#-javaHome#/scratch/oracle/jdk1.7.0_17#]
#####-archiveLoc#/tmp/a.jar]
#####-sourceDomainLoc#/scratch/oracle/config/domains/base_domain#]
#####-sourceOracleHomeLoc#/scratch/oracle/Oracle_home1/]
[]]
```

[[Table A-4] describes the options for the copyConfig script for standalone domains.]

Table A-5 [Options for the copyConfig Script for Standalone Domains

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-javaHome]	[NA]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[-archiveLoc]	[-al]	[The absolute path of the archive location. Use this option to specify the location of the archive file to be created by the copyConfig script.]	[Mandatory]
[-sourceDomainLoc]	[-sdl]	[The absolute path of the source domain containing the component.] [Note that on Windows, you should not include a backslash at the end of the path.]	[Mandatory]
[-sourceOracleHomeLoc]	[-soh]	[The absolute path of the source Oracle home.]	[Mandatory]

Table A-5 *[(Cont.)]Options for the copyConfig Script for Standalone Domains*

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
<code>[-logDirLoc]</code>	<code>[-ldl]</code>	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
<code>[-additionalParams]</code>	<code>[-ap]</code>	[An additional parameter and its value to be passed to the script. You must pass the variables as key=value pairs. Separate multiple variables with commas.] [On Windows, surround the entire value pair with double quotation marks (").]	[Optional]
<code>[-silent]</code>	<code>[NA]</code>	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of <code>[false]</code> . To continue, you must type <code>[yes]</code> , which is not case sensitive. Typing anything other than <code>yes</code> causes the script to abort.]	[Optional]

A.1.1.5 **[[copyConfig Script for Node Manager]**

[Creates a configuration archive that contains the snapshot of the configuration of Node Manager.]

[You must run the copyConfig script for each Node Manager in the source environment. A configuration archive is created for each source Node Manager.]

[The copyConfig script is located in:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/copyConfig.sh]
[ (Windows)#[ORACLE_COMMON_HOME]\bin\copyConfig.cmd]
[]]
```

[The syntax is:]

```
[[copyConfig##-javaHome#[path_of_jdk]]
[#####-archiveLoc#[archive_location]]
[#####-sourceNMHomeLoc#[source_Node_Manager_Home_location]]
[#####[-logDirLoc#[log_dir_path]]]
[#####[-silent#{true#|false}}]
[]]
```

[The following example shows how to create a copy of the source Node Manager configuration located in `/scratch/oracle/Oracle_home1/wlserver/common/nodemanager:`]

```
[[copyConfig.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/nm.jar]
[#####-sourceNMHomeLoc#/scratch/oracle/Oracle_
home1/wlserver/common/nodemanager]
[]]
```

[[Table A-6] describes the options for the copyConfig script for Node Manager.]

Table A-6 [Options for the copyConfig Script for Node Manager]

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[[-javaHome]]	[None]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[[-archiveLoc]]	[[-al]]	[The absolute path of the archive location. Use this option to specify the location of the archive file to be created by the copyConfig script.]	[Mandatory]
[[-sourceNMHomeLoc]]	[[-snh]]	[The absolute path of the source Node Manager home.]	[Mandatory]
[[-logDirLoc]]	[[-ldl]]	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
[[-silent]]	[None]	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of [false]. To continue, you must type [yes], which is not case sensitive. Typing anything other than yes causes the script to abort.]	[Optional]

A.1.1.6 [[extractMovePlan Script]

[Extracts configuration information from the archive into a move plan. It also extracts any needed configuration plans. Then, you edit the move plan, specifying properties for the target environment.]

[The extractMovePlan script is located in:]

```
[ (UNIX)#[ORACLE_COMMON_HOME]/bin/extractMovePlan.sh]
[ (Windows)#[ORACLE_COMMON_HOME]\bin\extractMovePlan.cmd]
[]
```

[The syntax is:]

```
[extractMovePlan#-javaHome#[path_of_jdk]]
[#####-archiveLoc#[archive_location]]
[#####-planDirLoc#[move_plan_directory]]
[##### [-optimizationHints#fusionApps,sameSchemaNameSinglePassword,#
rpdDataSource]]
[##### [-logDirLoc#[log_dir_path]]]
[]
```

[The following example extracts the plans from the archive j2ee.jar:]

```
[extractMovePlan.sh#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/j2ee.jar]
[#####-planDirLoc#/scratch/oracle/t2p_plans]
[]
```

[The extractMovePlan script extracts the move plan to the specified directory. Depending on the type of component that you are moving, the extractMovePlan script may also extract other configuration plans. For example, it might extract the following:]

```
[/scratch/oracle/t2p_plans/moveplan.xml]
[/scratch/oracle/t2p_plans/composites/configplan1.xml]
[/scratch/oracle/t2p_plans/composites/configplan2.xml]
[/scratch/oracle/t2p_plans/adapters/deploymentplan1.xml]
[/scratch/oracle/t2p_plans/adapters/deploymentplan2.xml]
[]
```

[[Table A-7] describes the options for the extractMovePlan script:]

Table A-7 []Options for the extractMovePlan Script

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-javaHome]	[NA]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[-archiveLoc]	[-al]	[The absolute path of the archive location. Use this option to specify the location of the archive file created by the copyConfig script.]	[Mandatory]
[-planDirLoc]	[-pdl]	[The absolute path to a directory to which the move plan, along with any needed configuration plans, is to be extracted.] [The directory must not exist or, if it exists, it is empty.]	[Mandatory]
[-optimizationHints]	[-opth]	[Specifies the configuration values to auto-populate based on the topology on the target environment. These values are omitted from the move plan. Use of the hints is recommended if they apply to your environment.] [This option takes the following arguments:] <ul style="list-style-type: none"> ▪ [[fusionapps (fa). If you use the -optimizationHints option, you must provide this hint.]] ▪ [[sameSchemaNameSinglePassword (ssnsp). The same password is used for all schemas associated with a particular database.]] ▪ [[rpdDataSource (rpdds). The same password is used for all schemas associated with a particular database specified in RPD_CONFIG section of the move plan.] [If the sameSchemaNameSinglePassword flag is provided, then this flag is automatically set to true.]]] 	[Optional]
[-logDirLoc]	[-ldl]	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]

[For information about the properties in the move plans, and which properties you should edit, see [Section A.2].]]

A.1.1.7 [[pasteConfig Script for Oracle WebLogic Server Domains]

[Applies the copied configurations from the source environment to the target environment. Inputs for the script include the location of the configuration archive created with the copyConfig script for the domain and the location of the modified move plan. The pasteConfig script re-creates the configuration information for the Oracle WebLogic Server domain in the target environment. It also merges the move plan property values for the target environment.]

[The pasteConfig script is located in:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/pasteConfig.sh]
[ (Windows)#[ORACLE_COMMON_HOME]\bin\pasteConfig.cmd]
[]]
```

[The syntax is:]

```
[[pasteConfig##-javaHome#[path_of_jdk]#]
[#####-archiveLoc#[archive_location]]
[#####-targetDomainLoc#[trgt_domain_path]#]
[#####-targetOracleHomeLoc#[trgt_Oracle_Home_path]]
```

```
[#####-movePlanLoc#[move_plan_path]#]
[#####-domainAdminPasswordFile#[domain_admin_password_file]#]
[#####[-appDir#[WLS_application_directory]]#]
[#####[-logDirLoc#[log_dir_path]]]
[#####[-silent#{true#|false}}]
[]]
```

[The following example shows how to apply the archive of the domain to the Oracle home Oracle_home1:]

```
[pasteConfig.sh#-javaHome#[scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#[tmp/java_ee_cl.jar]
[#####-targetDomainLoc#[scratch/oracle/config/domains/dom_cl]
[#####-targetOracleHomeLoc#[scratch/oracle/Oracle_home1#]
[#####-movePlanLoc#[scratch/oracle/java_ee/move_plan.xml]
[#####-domainAdminPasswordFile#[scratch/pwd_dir/pass.txt#]
[#####-logDirLoc#[tmp/log]]
```

[Note]: ■[[[If you are moving an environment that includes Oracle JRF, as in an Oracle Fusion Middleware Infrastructure installation, in the move plan, you must specify a different database for the target than is used in the source. The database host, port, service name, and schemas cannot be the same.]]]

[[Table A-8] describes the options for the pasteConfig script for Oracle WebLogic Server domains.]

Table A-8 []Options for the pasteConfig Script for Oracle WebLogic Server Domains

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-javaHome]	[NA]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[-archiveLoc]	[-al]	[The absolute path of the archive location. Use this option to specify the location of the archive file created by the copyConfig script.]	[Mandatory]
[-targetDomainLoc]	[-tdl]	[The absolute path of the target domain. The directory must not exist or, if it exists, it must be empty.] [The domain directory may be located outside of the directory structure of the Oracle home.]	[Mandatory]
[-targetOracleHomeLoc]	[-toh]	[The absolute path of the target Oracle home. It will be used to configure the domain.]	[Mandatory]
[-movePlanLoc]	[-mpl]	[The absolute path of the updated version of the move plan that was extracted from the source.]	[Mandatory]
[-domainAdminPassword File]	[-dap]	[The absolute path of a secure file containing the password for the administrative user for the domain on target environment. You must provide a password file, even if you are not changing the configuration.] [Note that the password is based on the authentication provider for the domain. For example, the authenticator can be an embedded LDAP or an external LDAP.]	[Mandatory.]

Table A-8 *[(Cont.)]Options for the pasteConfig Script for Oracle WebLogic Server Domains*

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-appDir]	[-ad]	[The absolute path of the Oracle WebLogic Server application directory on the target.]	[Optional]
[-logDirLoc]	[-ldl]	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
[-silent]	[NA]	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of [false]. To continue, you must type [yes], which is not case sensitive. Typing anything other than yes causes the script to abort.]	[Optional]

A.1.1.8 **[[pasteConfig Script for Standalone Domains]**

[Applies the copied configurations from the source environment to the target environment. Inputs for the script include the location of the configuration archive created with the copyConfig script for the domain and the location of the modified move plan. The pasteConfig script re-creates the configuration information for the standalone domain in the target environment. It also merges the move plan property values for the target environment.]

[The pasteConfig script is located in:]

```
[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/pasteConfig.sh]
[ (Windows)#[ORACLE_COMMON_HOME]\bin\pasteConfig.cmd]
[]]
```

[The syntax is:]

```
[[pasteConfig##-javaHome#[path_of_jdk]#]
[#####-archiveLoc#[archive_location]]
[#####-targetDomainLoc#[trgt_domain_path]#]
[#####-targetOracleHomeLoc#[trgt_Oracle_Home_path]]
[#####-movePlanLoc#[move_plan_path]#]
[#####[-logDirLoc#[log_dir_path]]]
[#####[-silent#{true#|#false}]]
[]]
```

[The following example shows how to apply the archive of the domain to the Oracle home Oracle_home1:]

```
[[pasteConfig.sh##-javaHome#/scratch/oracle/jdk1.7.7.0_17]
[#####-archiveLoc#/tmp/java_ee_cl.jar]
[#####-targetDomainLoc#/scratch/oracle/config/domains/dom_cl]
[#####-targetOracleHomeLoc#/scratch/oracle/Oracle_home1#]
[#####-movePlanLoc#/scratch/oracle/java_ee/move_plan.xml]
[#####-logDirLoc#/tmp/log]
[]]
```

[[Table A-8] describes the options for the pasteConfig script for standalone domains.]

Table A–9 []Options for the pasteConfig Script for Standalone Domains

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[-javaHome]	[NA]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[-archiveLoc]	[-al]	[The absolute path of the archive location. Use this option to specify the location of the archive file created by the copyConfig script.]	[Mandatory]
[-targetDomainLoc]	[-tdl]	[The absolute path of the target domain. The directory must not exist or, if it exists, it must be empty.] [The domain directory may be located outside of the directory structure of the Oracle home.]	[Mandatory]
[-targetOracleHomeLoc]	[-toh]	[The absolute path of the target Oracle home. It will be used to configure the domain.]	[Mandatory]
[-movePlanLoc]	[-mpl]	[The absolute path of the updated version of the move plan that was extracted from the source.]	[Mandatory]
[-logDirLoc]	[-ldl]	[The location of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
[-silent]	[NA]	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of [false]. To continue, you must type [yes], which is not case sensitive. Typing anything other than yes causes the script to abort.]	[Optional]

A.1.1.9 [[pasteConfig Script for Node Manager]

[Applies the copied configurations of Node Manager from the source environment to the target environment. Inputs for the script include the location of the configuration archive created with the copyConfig script for the Oracle WebLogic Server Node Manager and the location of the modified move plan. The pasteConfig script re-creates the configuration information for Node Manager in the target environment. It also merges the move plan property values for the target environment.]

[Note]: [All the domains that are to be managed by Node Manager should be moved before applying the copy of Node Manager to the target environment. In addition, the Administration Server must be running.]

[You must run the pasteConfig script for each Node Manager in the target environment.]

[The syntax is:]

```
[ [pasteConfig##-javaHome# [path_of_jdk]# ]
[#####-archiveLoc# [archive_location] ]
[#####-targetNMHomeLoc# [trgt_Node_Manager_Home_path]# ]
[#####-targetOracleHomeLoc# [trgt_Oracle_Home_path]# ]
[#####-movePlanLoc# [move_plan_path] ]
[##### [-logDirLoc# [log_dir_path] ] ]
```

```

#####[-silent#{true#|#false}]
[]
[The following example shows how to apply the copy of Node Manager to the Node
Manager home located in /scratch/Oracle_
home1/wlserver/common/nodemanager:]

[[pasteConfig#-javaHome#/scratch/oracle/jdk1.7.7.0_17]
#####-archiveLoc#/tmp/nm.jar]
#####-targetNMHomeLoc#/scratch/oracle/Oracle_
home1/wlserver/common/nodemanager]
#####-targetOracleHomeLoc#/scratch/oracle/Oracle_home1]
#####-movePlanLoc#/scratch/oracle/t2ppplans/nm/moveplan.xml]
[]

```

[[Table A-10] describes the options for the pasteConfig script for Node Manager.]

Table A-10 []Options for the pasteConfig Script for Node Manager

[Options]	[Shortcut]	[Description]	[Mandatory or Optional]
[[-javaHome]]	[None]	[The absolute path of the Java Developer's Kit.] [The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.]	[Mandatory]
[[-archiveLoc]]	[[-al]]	[The absolute path of the archive location. Use this option to specify the location of the archive file created by the copyConfig script.]	[Mandatory]
[[-targetNMHomeLoc]]	[[-tnh]]	[The absolute path of the target Node Manager.]	[Mandatory]
[[-targetOracleHomeLoc]]	[[-tohc]]	[The absolute path of the target Oracle home that will be used to configure Node Manager.]	[Mandatory]
[[-movePlanLoc]]	[[-mpl]]	[The absolute path of the modified move plan in the target environment.]	[Mandatory]
[[-logDirLoc]]	[[-ldl]]	[The absolute path of a directory. The directory may or may not exist. A new log file is created in the directory. The default is the system Temp location.]	[Optional]
[[-silent]]	[None]	[Specifies whether the operation operates silently. That is, it does not prompt for confirmation. The default is that the operation does not prompt for confirmation.] [To specify that it does prompt for confirmation, specify this option with the value of [false]. To continue, you must type [yes], which is not case sensitive. Typing anything other than yes causes the script to abort.]	[Optional]

A.1.1.10 [[obfuscatePassword Script]

[Generates a file that contains the obfuscated password. In the scripts and in the move plans, you often need to provide files containing passwords. The script prompts you to enter the password and the location where the password file is to be written.]

[The syntax is:]

```

[[ (UNIX)#[ORACLE_COMMON_HOME]/bin/obfuscatePassword.sh]
[[ (Windows)#[ORACLE_COMMON_HOME]\bin\obfuscatePassword.cmd]]]

```

A.2 [[Modifying Move Plans]

[When you move Oracle Fusion Middleware components, you run the extractMovePlan script to create a move plan for the entity that you are moving. The extractMovePlan script extracts configuration information from the archive into a

move plan. It also extracts any needed configuration plans. Before you apply the archive to the target, you must edit the move plan to reflect the values of the target environment.]

[You can modify properties with the scope of READ_WRITE. Do not modify the properties with the scope of READ_ONLY.]

[Notes]: []

- [[Do not add, comment, or remove any section of a move plan.]]
 - [[You must generate a move plan using the extractMovePlan script each time you create an archive, even if you have made no changes to the move plan. If you use a move plan that was created with a different archive, the archive will not be accepted and the pasteConfig script will fail.]]
-
-

[This section provides the following topics:]

- [\[\[Locating configGroup Elements\]\]](#)
- [\[\[Move Plan Properties\]\]](#)

A.2.1 [[Locating configGroup Elements]

[Most move plans contain multiple configGroup elements. When a property is associated with a particular configGroup element, the tables listing the properties group the properties by configGroup element. For example, [Table A-13], which shows the properties for the move plan for Java components, shows multiple configGroup elements, such as SERVER_CONFIG and MACHINE_CONFIG.]

[The following example shows a portion of the move plan for Java components, with portions of the SERVER_CONFIG and MACHINE_CONFIG configGroup elements:]

```
[ [<?xml#version="1.0"#encoding="UTF-8"#standalone="yes" ?>]
[<movePlan>]
[#####<movableComponent>]
[#####<componentType>J2EEDomain</componentType>]
[#####<moveDescriptor>]
[#####<StartupMode>PRODUCTION</StartupMode>]
[#####<configGroup>]
[#####<type>SERVER_CONFIG</type>]
[#####<configProperty#id="Server1">]
[#####<configProperty>]
[#####<name>Server#Name</name>]
[#####<value>AdminServer</value>]
[#####<itemMetadata>]
[#####<dataType>STRING</dataType>]
[#####<scope>READ_ONLY</scope>]
[#####</itemMetadata>]
[#####</configProperty>]
[#####.]
[#####.]
[#####.]
[#####</configGroup>]
[#####<configGroup>]
[#####<type>MACHINE_CONFIG</type>]
[#####<configProperty#id="Machine1">]
[#####<configProperty>]
[#####<name>Machine#Name</name>]
```

```

[#####<value>LocalMachine</value>]
[#####<itemMetadata>]
[#####<dataType>STRING</dataType>]
[#####<scope>READ_WRITE</scope>]
[#####</itemMetadata>]
[#####</configProperty>]
[#####<configProperty>]
[#####<name>Node#Manager#Listen#Address</name>]
[#####<value>example.com</value>]
[#####<itemMetadata>]
[#####<dataType>STRING</dataType>]
[#####<scope>READ_WRITE</scope>]
[#####</itemMetadata>]
[#####</configProperty>]
[#####. ]
[#####. ]
[#####. ]
[#####</configGroup>]]]

```

A.2.2 [[Move Plan Properties]

[The tables in this section describe the move plan properties you can customize for Oracle Fusion Middleware entities and components.]

"

[Note]: [Many move plan properties require that you provide the location of a file containing a password. If you want to use obfuscated passwords, you can use the `obfuscatePassword` script, as described in [\[Section A.1.1.10\]](#).]

[The properties that you edit differ depending on the type of component. [\[Table A–11\]](#) provides pointers to the appropriate list of properties for each component.]

Table A–11 []Move Plan Properties for Components

[Component]	[Where to find the list of properties:]
[Node Manager]	[[Table A–12]]
[All Java components]	[[Table A–13]]
[Oracle ADF connections]	[[Table A–14]]
[Oracle Coherence]	[[Table A–15]]
[Oracle HTTP Server]	[[Table A–17]]
[Oracle Web Services Manager]	[[Table A–18]]
[Oracle Data Integrator]	[[Table A–19]]
[Oracle User Messaging Service]	[[Table A–13], [Table A–14], [Table A–16]]

[[[Table A–12](#)]] describes the move plan properties that you can change for Node Manager. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment.]

Table A–12 []Move Plan Properties for Node Manager

[Property]	[Description]	[Sample Value]
[[Properties in the NODE_MANAGER_PROPERTIES configGroup:]]	[Node Manager configuration]	[]
[Node Manager Home]	[The absolute location of Node Manager]	[/scratch/oracle/domains/base_domain/nodemanager]
[Listen Address]	[The Listen address of Node Manager.]	[[example.com]]
[Listen Port]	[The number of the Listen port of Node Manager.]	[[5557]]
[Custom Identity KeyStore File]	[The absolute path of the custom identity keystore file location.] [This property is present in the move plan only if the source environment is configured with SSL.]	[[/scratch/oracle/Oracle_home1/wlserver/server/lib/example_identity.jks]]
[Custom Identity Private Key Alias]	[The value of the identity key store alias.] [This property is present in the move plan only if the source environment is configured with SSL.]	[[mykey]]
[Custom Identity Private Key Passphrase File]	[The absolute path to the secure file containing the private key used when creating a certificate.] [This property is present in the move plan only if the source environment is configured with SSL.]	[[/scratch/oracle/key_passwd]]
[Custom Identity Keystore Passphrase File]	[The absolute path to the secure file containing the custom identity keystore password.] [If source environment uses DemoTrust, these properties are optional. If left blank, the pasteConfig script shows a warning message and proceed. If source environment uses Custom, these parameters are mandatory.]	[[/scratch/oracle/i_passwd.txt]]
[Java Standard Trust Keystore Passphrase File]	[The absolute path of the Java standard trust keystore file location.]	[]
[]	[]	[]
[[Properties in the DOMAINS configGroup:]]	[Oracle WebLogic Server domain configuration]	[]
[Domain Name]	[The name of the domain.]	[[WLS_domain]]
[Domain Location]	[The absolute path of the domain location.]	[[/scratch/oracle/config/domains/WLS_domain]]
[AdminServer Listen Address]	[The Listen address of the Administration Server.]	[[example.com]]

Table A–12 [(Cont.)]*Move Plan Properties for Node Manager*

[Property]	[Description]	[Sample Value]
[AdminServer Listen Port]	[The number of the Listen port of the Administration Server.]	[[7001]]
[AdminServer User Name]	[The administration user name.]	[[weblogic]]
[AdminServer Password]	[The absolute path to the secure file containing the administration user's password.]	[[/scratch/oracle/admin_passwd]]
[Node Manager User Name]	[The Node Manager user name.]	[[weblogic]]
[Node Manager Password]	[The absolute path to the secure file containing the Node Manager user's password.]	[[/scratch/oracle/nm_passwd]]

[[Table A–13] describes the move plan properties that you can change for Java components. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment.]

Table A–13 []*Common Move Plan Properties for Java Components*

[Property]	[Description]	[Sample Value]
[Startup Mode]	[The startup mode of an Oracle WebLogic Server domain.] [Valid values are:] <ul style="list-style-type: none"> ▪ [[DEVELOPMENT. Use this mode while you are developing your applications. Development mode uses a relaxed security configuration and enables you to auto-deploy applications.]] ▪ [[PRODUCTION. Use this mode when your application is running in its final form. A production domain uses full security and may use clusters or other advanced features.]]] [The default is PRODUCTION.]	[[PRODUCTION]]
[[Properties in the SERVER_CONFIG configGroup:]]	[Common Java component properties]	[]
[Administration Port]	[The port number for the Administration Server.]	[[7001]]
[Listen Address]	[The Listen address of the WebLogic Server. Set it to the host name or set it to [All#Local#Addresses] to listen on all addresses on the host.]	[[All#Local#Addresses]]
[Listen Port]	[The number of the Listen port.] [If you do not provide a port number or if the port number you provide is not available, the operation returns an error.]	[[8001]]
[SSL Listen Port]	[The number of the SSL Listen port. This property is present in the move plan if SSL is enabled.]	[[7002]]

Table A–13 [(Cont.)]Common Move Plan Properties for Java Components

[Property]	[Description]	[Sample Value]
[Frontend Host]	[The host name of the HTTP Server.] [This property is present in the move plan only if the HTTP Server is set as the frontend to the server.]	[[example.com]]
[Frontend HTTP Port]	[The number of the HTTP Server port.] [This property is present in the move plan only if the HTTP Server is set as the frontend to the server.]	[[10605]]
[Frontend HTTPS Port]	[The number of the HTTPS Server port.] [This property is present in the move plan only if the HTTP Server is set as the frontend to the server.]	[[10666]]
[KeyStore Type]	[The type of keystore.]	[[JKS]]
[Custom Identity Keystore File]	[The absolute path of custom identity keystore file location.] [If the Startup Mode is DEVELOPMENT and the source environment uses DemoTrust as the keystore, you can leave this parameter blank. The same type of store will be created in the target environment.] [If the Startup Mode is DEVELOPMENT and the source environment uses CustomTrust as the keystore, you must provide a value for this parameter. The same type of store will be created in the target environment.] [If the Startup Mode is PRODUCTION and SSL is enabled in the source environment (regardless of whether you use DemoTrust or CustomTrust), you must provide a value for this parameter.]	[[/scratch/oracle/Oracle_home1/wlserver/server/lib/example_identity.jks]]
[Custom Identity Keystore Passphrase File]	[The absolute path to the secure file containing the custom identity keystore password.] [If the Startup Mode is DEVELOPMENT and the source environment uses DemoTrust as the keystore, you can leave this parameter blank. The same type of store will be created in the target environment.] [If the Startup Mode is DEVELOPMENT and the source environment uses CustomTrust as the keystore, you must provide a value for this parameter.] [If the Startup Mode is PRODUCTION and SSL is enabled in the source environment (regardless of whether you use DemoTrust or CustomTrust), you must provide a value for this parameter.]	[[/scratch/oracle/i_passwd]]

Table A-13 [(Cont.)]Common Move Plan Properties for Java Components

[Property]	[Description]	[Sample Value]
[Custom Trust Keystore File]	<p>[The absolute path of custom trust keystore file location.]</p> <p>[If the Startup Mode is DEVELOPMENT and the source environment uses DemoTrust as the keystore, you can leave this parameter blank. The same type of store will be created in the target environment.]</p> <p>[If the Startup Mode is DEVELOPMENT and the source environment uses CustomTrust as the keystore, you must provide a value for this parameter.]</p> <p>[If the Startup Mode is PRODUCTION and SSL is enabled in the source environment (regardless of whether you use DemoTrust or CustomTrust), you must provide a value for this parameter.]</p>	[[/scratch/Oracle/Middleware/wlserver/server/lib/example_trust.jks]]
[Custom Trust Keystore Passphrase File]	<p>[The absolute path to the secure file containing the custom trust keystore password.]</p> <p>[If the Startup Mode is DEVELOPMENT and the source environment uses DemoTrust as the keystore, you can leave this parameter blank. The same type of store will be created in the target environment.]</p> <p>[If the Startup Mode is DEVELOPMENT and the source environment uses CustomTrust as the keystore, you must provide a value for this parameter.]</p> <p>[If the Startup Mode is PRODUCTION and SSL is enabled in the source environment (regardless of whether you use DemoTrust or CustomTrust), you must provide a value for this parameter.]</p>	[[/scratch/oracle/key_passwd]]
[Custom Identity Private Key Alias]	<p>[The private key alias.]</p> <p>[If the Startup Mode is DEVELOPMENT and the source environment uses DemoTrust as the keystore, you can leave this parameter blank. The same type of store will be created in the target environment.]</p> <p>[If the Startup Mode is DEVELOPMENT and the source environment uses CustomTrust as the keystore, you must provide a value for this parameter.]</p> <p>[If the Startup Mode is PRODUCTION and SSL is enabled in the source environment (regardless of whether you use DemoTrust or CustomTrust), you must provide a value for this parameter.]</p>	[[fmw_key]]

Table A-13 [(Cont.)]Common Move Plan Properties for Java Components

[Property]	[Description]	[Sample Value]
[Custom Identity Private Key Passphrase File]	[The absolute path to the secure file containing the custom identity private key password.] [If the Startup Mode is DEVELOPMENT and the source environment uses DemoTrust as the keystore, you can leave this parameter blank. The same type of store will be created in the target environment.] [If the Startup Mode is DEVELOPMENT and the source environment uses CustomTrust as the keystore, you must provide a value for this parameter.]	[[/scratch/oracle/i_passwd]]
[[Properties in the CLUSTER_CONFIG configGroup:]]	[Oracle WebLogic Server Cluster configuration properties]	[]
[Messaging Mode]	[The cluster messaging mode. Acceptable values are [unicast] and [multicast].]	[[multicast]]
[Cluster Address]	[The cluster address.]	[[localhost]]
[Unicast Channel]	[The name of the unicast channel.]	[[MyMulticastChannel]]
[Multicast Address]	[The multicast address.]	[[239.192.0.0]]
[Multicast Port]	[The port number of the multicast address.]	[[8899]]
[Frontend Host]	[The name or IP address of the front-end host for the cluster.]	[[example.com]]
[Frontend HTTP Port]	[The HTTP port number for the front-end host for the cluster.]	[[7008]]
[Frontend HTTPS Port]	[The HTTPS port number for the front-end host for the cluster.]	[[7009]]
[[Properties in the MACHINE_CONFIG configGroup:]]	[Machine configuration properties]	[]
[Machine Name]	[The name of the machine.]	[[example.com]]
[Node Manager Listen Address]	[The Listen address of the machine running Node Manager.]	[[example.com]]
[Node Manager Listen Port]	[The port number of the Listen address of the machine running Node Manager.]	[[5556]]
[[Property in the DEPLOYMENT_PLAN_CONFIG configGroup:]]	[Deployment plans]	[]
[Application Name]	[The name of the application.]	[[helloWorldEar]]
[Deployment Plan]	[The location where an application's deployment plan is to be extracted. The location is relative to the location of the move plan.]	[[deploy_plans/helloWorldEar_plan.xml]]
[[Properties in the AUTHENTICATORS configGroup:]]	[Authenticator configuration]	[]
[Host Name]	[The LDAP server host name.]	[[example.com]]
[Port]	[The LDAP server port number.]	[[3060]]
[Principal]	[The administration user for the LDAP server.]	[[cn=orcladmin]]

Table A–13 [(Cont.)]Common Move Plan Properties for Java Components

[Property]	[Description]	[Sample Value]
[Password File]	[The absolute path of a secure file containing the password for the LDAP user. You must provide a password file, even if you are not changing the configuration.]	[[/scratch/oracle/ldap_passwd.txt]]
[User Base DN]	[The user base distinguished name (DN).]	[[cn=users,dc=us,dc=oracle,dc=com#]]
[User Object Class]	[The user object class.]	[[person]]
[Group Base DN]	[The group base distinguished name (DN).]	[[cn=groups,dc=us,dc=oracle,dc=com#]]
[GUID Attribute]	[The global unique identifier.]	[[orclguid]]
[[Properties in the DATASOURCE configGroup:]]	[Data source configuration]	[]
[Driver Class]	[The driver class of the data source. Refer to "Using JDBC Drivers with WebLogic Server" in the <i>[Administering JDBC Data Sources for Oracle WebLogic Server]</i> to choose the appropriate class.]	[[oracle.jdbc.OracleDriver]]
[Url]	[The URL of the database for the data source. It contains the host name, the database port number, and SID.] [It has the following format:] [[jdbc:oracle:thin:@[Db_host]:[Db_port]:[Db_SID]#]]	[[jdbc:oracle:thin:@host.examplecom:1521:orcl]]
[User]	[The schema name of the data source.]	[[OFM_MDS]]
[Password File]	[The absolute path to the secure file containing the password of the database schema. You must provide a password file, even if you are not changing the configuration of the data source.]	[[/scratch/oracle/ds_passwd.txt]]
[ONS Node List]	[The list of Oracle Notification Service (ONS) hosts and ports, in the following format:] [[[ons_host1:port1], [ons_host2:port2]]]	[[myhost1:6100,myhost2:6101]]
[ONS Wallet File]	[The absolute path to the credential store file, which contains keys and certificates. This property is configured only if SSL is enabled.]	[[/scratch/wallet]]
[ONS Wallet Password File]	[The absolute path to the file containing the password for the wallet.]	[[/scratch/ons_pass.txt]]

Table A-13 [(Cont.)]Common Move Plan Properties for Java Components

[Property]	[Description]	[Sample Value]
[[Properties in the OPSS_SECURITY configGroup, in the configProperty with the ID of LDAP:]]	<p>[LDAP-based policy and credential store configuration.]</p> <p>[If the source is a file-based store, these properties, as well as the LDAP-based and database-based Policy and Credential Store properties are present in the move plan. When you configure the move plan, you can change from a file-based to an LDAP or database-based store.]</p> <p>[If the source is LDAP-based, only the LDAP properties are present in the move plan. You cannot change it to a different type, but you can change the LDAP endpoints.]</p> <p>[If the source is database-based, only the database properties are present in the move plan. You cannot change it to a different type, but you can change the database-based endpoints.]</p> <p>[You can only use one type of store. To use one, uncomment the section in the move plan and ensure the other is commented.]</p>	[]
[Password File]	[The absolute path to the secure file containing the password of the LDAP Server Administrative user. You must provide a password file, even if you are not changing the configuration of the LDAP Server.]	[/scratch/oracle/ldap_passwd.txt]
[LDAP User]	[The LDAP Server administrative user name.]	[[cn=orcladmin]]
[jps Root]	[The LDAP Server context root.]	[[cn=jpsRoot]]
[Domain]	[The name of the domain.]	[[WLS_domain]]
[LDAP Url]	[The URL of the LDAP connection. It contains the host name and port number of the LDAP store.]	[[ldap://example.com:3060]]
[[Properties in the OPSS_SECURITY configGroup, in the configProperty with the ID of DB:]]	<p>[Database-based policy and credential store configuration.]</p> <p>[If the source is a file-based store, these properties are present in the move plan. (The LDAP-based store is not present and you cannot move from a database-based to an LDAP-based store.) When you configure the move plan, you can change from a file-based to an LDAP or database-based store.]</p> <p>[You can only use one type of store. To use one, uncomment the section in the move plan and ensure the other is commented.]</p> <p>[If the source is database-based, only the database properties are present in the move plan. You cannot change it to a different type, but you can change the database-based endpoints.]</p>	[]
[Password File]	[The absolute path to the secure file containing the password of the OPSS schema owner. You must provide a password file, even if you are not changing the configuration.]	[/scratch/oracle/ldap_passwd.txt]

Table A-13 [(Cont.)]Common Move Plan Properties for Java Components

[Property]	[Description]	[Sample Value]
[DataSource Name]	[The name of the data source. The name cannot contain a slash (/).]	[[opssds]]
[DataSource Jndi Name]	[The JNDI name of the data source.]	[[jdbc/opss]]
[Jps Root]	[The LDAP Server context root.]	[[cn=jpsRoot]]
[Domain]	[The name of the domain.]	[[WLS_domain]]
[Driver Class]	[The driver class of the data source. Refer to ["Selecting a JDBC Driver"] in the <i>[Administering JDBC Data Sources for Oracle WebLogic Server]</i> to choose the appropriate class.]	[[oracle.jdbc.OracleDriver]]
[Url]	[The URL of the database for the data source. It contains the host name, the database port number, and SID.] [It has the following format:] [[jdbc:oracle:thin:@[Db_host]:[Db_port]:[Db_SID]]]	[[jdbc:oracle:thin:@host.example.com:1521:orcl]]
[User]	[The name of the OPSS schema owner of the data source.]	[[DEV_OPSS]]
[[Properties in the RDBMS Security Store configGroup:]]	[Database-based security store configuration]	[[]]
[URL]	[The URL of the database for the data source. It contains the host name, the database port number, and SID.] [It has the following format:] [[jdbc:oracle:thin:@[Db_host]:[Db_port]:[Db_SID]#]]	[[jdbc:oracle:thin:@host.example.com:1521:orcl]]
[Driver Class]	[The driver class of the RDBMS Security Store connection. Refer to ["Using JDBC Drivers with WebLogic Server"] in the <i>[Administering JDBC Data Sources for Oracle WebLogic Server]</i> to choose the appropriate class.]	[[oracle.jdbc.OracleDriver]]
[User]	[The name of the schema owner.]	[[admin]]
[Password File]	[The absolute path to the secure file containing the password of the security store schema owner. You must provide a password file, even if you are not changing the configuration.]	[[/scratch/oracle/rbms_passwd.txt]]
[[Property in the ADAPTER configGroup:]]	[Resource adapter configuration]	[[]]
[Deployment Plan]	[The path to the deployment plan to be used during movement to the target. The path can be absolute, or relative to the location of the move plan.] [The deployment plan is extracted by the extractMovePlan script.]	[[/scratch/adapters/adapters.xml]]

[[Table A-14] describes the move plan properties that you can change if you are using Oracle ADF connections. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment. The table is divided by

component. For some components, the description column lists the OBJECT_NAME_PROPERTY type. You can search for the type to locate the relevant section.]

Table A-14 []Move Plan Properties for Oracle ADF Connections

[Property]	[Description]	[Sample Value]
[[Oracle ADF URL Connection]]	[[OBJECT_NAME_PROPERTY type is URLConnProvider]]	[[]]
[Port]	[The port number used for the URL connections.]	[[7000]]
[URL]	[The URL used for the connection.]	[[example.com]]
[[Oracle ADF Business Components Service Connection]]	[[OBJECT_NAME_PROPERTY type is ADFBCServiceConnection]]	[[]]
[ServiceEndpointProvider]	[The Business Components service endpoint provider.]	[[ADFBC]]
[JndiFactoryInitial]	[The JNDI initial factory class.]	[[com.sun.java.jndi.InitialFactory]]
[JndiProviderUrl]	[The URL of the JNDI provider.]	[[t3://example.com:7101]]
[JndiSecurityPrincipal]	[The JNDI security principal name.]	[[weblogic]]
[WebServiceConnectionName]	[The Web service connection name.]	[[test]]
[[Oracle Enterprise Scheduler]]	[[OBJECT_NAME_PROPERTY type is EssConnection]]	[[]]
[NotificationServiceURL]	[The Oracle Enterprise Scheduler notification service URL.]	[[http://localhost:8001]]
[RequestFileDirectory]	[The path of the directory where request logs for jobs from OES ConcurrentProcessor (CP) extension is to be created.]	[[/tmp/ess/requestFileDirectory]]
[SAMLTokenPolicyURI]	[The SAML Policy URI to be used by CP extension.]	[[oracle/wss11_saml_token_with_message_protection_service_policy]]
[EssCallbackClientSecurityPolicyURI]	[The security policy to be used in the WS-Security headers for Web service invocations from Oracle Enterprise Scheduler for Web service callbacks.]	[[oracle/wss11_saml_token_with_message_protection_client_policy]]
[[Oracle Essbase]]	[[OBJECT_NAME_PROPERTY type is EssbaseConnProvider]]	[[]]
[Host]	[The host name for the Oracle Essbase server.]	[[example.com]]
[Cluster]	[The name of the cluster of which the Oracle Essbase server is a member.]	[[esbCluster]]
[Port]	[The Listen port number of the Oracle Essbase server.]	[[1423]]
[Username]	[The user name.]	[[user3]]
[[Oracle Web Services]]	[[OBJECT_NAME_PROPERTY type is WebServiceConnection]]	[[]]
[WsdUrl]	[The URL for the WSDL.]	[[http://example.com:[port]/MyWebService1?WSDL]]

Table A-14 *[(Cont.)]Move Plan Properties for Oracle ADF Connections*

[Property]	[Description]	[Sample Value]
[[Oracle Web Services]]	[[OBJECT_NAME_PROPERTY type is Port]]	[[{}]]
[AddressUrl]	[The service endpoint URL.]	[[http://example.com: [port]/MyWebService1]]
[ProxyHost]	[The name of the host on which the proxy server is running.]	[[example.com]]
[ProxyPort]	[The port number to which the proxy server is listening.]	[[80]]

[[Table A-15] describes the move plan properties that you can change for Oracle Coherence. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment.]

Table A-15 *[]Move Plan Properties for Oracle Coherence*

[Property]	[Description]	[Example]
[[Properties in the COHERENCE_SERVER_CONFIG configGroup]]	[The configuration of Oracle Coherence servers.]	[[{}]]
[Unicast Listen Address]	[The unicast listen address. This property is not present in the move plan if the value is localhost or a loopback address or the value is empty.]	[[10.240.98.232]]
[Unicast Listen Port]	[The unicast listen port.]	[[7036]]
[[Properties in the COHERENCE_CLUSTER_CONFIG configGroup]]	[The configuration of Oracle Coherence clusters.]	[[{}]]
[Custom Cluster Configuration File]	[An external custom configuration file used to configure this cluster.]	[[/scratch/external_custom_config.xml]]
[Unicast Listen Address]	[The unicast listen address. If the clustering mode is unicast, the properties Unicast Listen Address and Unicast Listen Port are present in the move plan. However, This property is not present in the move plan if the value of Unicast Listen Address is localhost or a loopback address or the value is empty.]	[[10.240.98.232]]
[Unicast Listen Port]	[The unicast listen port. This property is present in the move plan if the clustering mode is unicast.]	[[7036]]
[Multicast Listen Address]	[The multicast listen address. If the clustering mode is not unicast, this property is present in the move plan.]	[224.12.1.0]
[Multicast Listen Port]	[The multicast listen address. This property is present in the move plan if the clustering mode is not unicast.]	[12100]
[Listen Address]	[The listen address for the Well Known Address Name configuration property.]	[[{}]]
[Listen Port]	[The listen port for the Well Known Address Name configuration property]	[[{}]]

Table A-15 [(Cont.)]Move Plan Properties for Oracle Coherence

[Property]	[Description]	[Example]
[[Properties in the SERVER_TEMPLATES_CONFIG configGroup:]]	[Server templates configuration properties. These properties are visible in the move plan if the Coherence cluster configuration is overridden in server templates.]	[]
[Unicast Listen Address]	[The unicast listen address.]	[[10.240.98.232]]
[Unicast Listen Port]	[The unicast listen port.]	[[7036]]

[[Table A-16] describes the move plan properties that you can change for Oracle User Messaging Service. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment.]

Table A-16 []Move Plan Properties for Oracle User Messaging Service

[Property]	[Description]	[Sample Value]
[[Properties for the componentType UMS:]]	[These properties apply to all driver types.]	[[[]]]
[DefaultSenderAddress]	[The default address of the sender.]	[[SMS:1234]]
[SenderAddresses]	[The list of sender addresses that the driver supports.]	[[SMS:1234]]
[[Properties in the EmailDriver componentType:]]	[These properties apply to email drivers.]	[[[]]]
[IncomingMailServer]	[The host name of the incoming mail server.]	[[usmdemo.example.com]]
[IncomingMailServerPort]	[Port number of IMAP4 (that is, 143 or 993) or POP3 (that is, 110 or 995) server.]	[[110]]
[IncomingUserIDs]	[The list of user names of the mail accounts from which the driver instance is polling. Each name must be separated by a comma, for example, user1,user2.]	[[[username].pop3@usmdemo.example.com]]
[IncomingMailIDs]	[The email addresses corresponding to the user names. Each email address is separated by a comma and must reside in the same position in the list as its corresponding user name appears on the usernames list.]	[[[username].pop3@usmdemo.example.com]]
[IncomingUserPasswords]	[The absolute path of a secure file containing the password. You must provide a password file, even if you are not changing the configuration.] [The file contains a list of passwords corresponding to the user names. Each password is separated by a comma and must reside in the same position in the list as their corresponding user name appears on the usernames list.]	[[/scratch/oracle/ums_in_passwd.txt]]
[OutgoingMailServer]	[The name of the SMTP server.]	[[usmdemo.example.com]]
[OutgoingMailServerPort]	[The port number of the SMTP server.]	[[25]]
[OutgoingDefaultFromAddr]	[The default FROM address (if one is not provided in the outgoing message).]	[[[username].pop3@usmdemo.example.com]]
[OutgoingUsername]	[The user name used for SMTP authentication.]	[[[username].pop3@usmdemo.example.com]]

Table A-16 *[(Cont.)]Move Plan Properties for Oracle User Messaging Service*

[Property]	[Description]	[Sample Value]
[OutgoingPassword]	[The absolute path of a secure file containing the password used for SMTP authentication. This is required only if SMTP authentication is supported by the SMTP server. You must provide a password file, even if you are not changing the configuration.] [The file includes the type of password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, and Use Cleartext Password) and Password.]	[[/scratch/oracle/ums_out_passwd.txt]]
[[Properties in the ExtensionEndpoint configGroup:]]	[These properties apply to extension drivers.]	[[[]]]
[EndpointURL]	[Remote endpoint listener URL]	[[http://hostname:7001/integrationtest-war/extension]]
[mappedDomain]	[The extension endpoint used to deliver messages where the domain part of the recipient URI matches this value.]	[[test]]
[[Properties in the SMPPDriver componentType:]]	[These properties apply to SMPP drivers.]	[[[]]]
[SmsAccountId]	[The Account Identifier on the SMS-C.]	[[myusername]]
[SmsServerHost]	[The name (or IP address) of the SMS-C server.]	[[example_host]]
[TransmitterSystemId]	[The account ID that is used to send messages.]	[[myusername]]
[ReceiverSystemId]	[The account ID that is used to receive messages.]	[[myusername]]
[TransmitterSystemType]	[The type of transmitter system.]	[[Logica]]
[ReceiverSystemType]	[The type of receiver system.]	[[Logica]]
[ServerTransmitterPort]	[The TCP port number of the transmitter server.]	[[9001]]
[ServerReceiverPort]	[The TCP port number of the receiver server.]	[[9001]]
[TransmitterSystemPassword]	[The absolute path of a secure file containing the password of the transmitter system You must provide a password file, even if you are not changing the configuration.] [The file contains the type of password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, and Use Cleartext Password) and Password.]	[[/scratch/oracle/ums_trans_passwd.txt]]
[ReceiverSystemPassword]	[The absolute path of a secure file containing the password of the receiver system You must provide a password file, even if you are not changing the configuration.] [The file contains the type of password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, and Use Cleartext Password) and Password.]	[[/scratch/oracle/ums_rec_passwd.txt]]
[[Property in the WorklistDriver componentType:]]	[This property applies to the Worklist drivers.]	[[[]]]
[BPELConnectionURL]	[The URL of the BPEL server to connect to. The format is 'http://[bpel-host]:[bpel-port]'. The default behavior, if this property is not set, is to use the local container's HTTP connection URL.]	[[http://host.example.com:[port]]]

Table A–16 [(Cont.)]Move Plan Properties for Oracle User Messaging Service

[Property]	[Description]	[Sample Value]
[[Properties in the XMPPDriver componentType:]]	[These properties apply to XMPP drivers.]	[[]]
[IMServerHost]	[The Jabber/XMPP server host name.]	[[example.domain.com]]
[IMServerPort]	[The corresponding Jabber/XMPP server port. The default is 5222.]	[[5222]]
[IMServerUsername]	[The Jabber/XMPP user name with which you log in. You may also enter a complete Jabber ID if its domain name is different from the Jabber/XMPP server host name.]	[[myUserName@[xmpp-domain]]]
[IMServerPassword]	[The absolute path of a secure file containing the corresponding password for the IMServerUsername. You must provide a password file, even if you are not changing the configuration.] [The file contains the type of password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, Use Cleartext Password) and Password.]	[[/scratch/oracle/ums_im_passwd.txt]]

[[Table A–17] describes the move plan properties that you can change for Oracle HTTP Server. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment.]

[For Oracle HTTP Server, there are many configGroup elements in the move plan. Each configGroup element is associated with one Oracle HTTP Server configuration file. As a result, there may be more than one instance of a particular property, such as User.]

Table A–17 []Move Plan Properties for Oracle HTTP Server

[Property]	[Description]	[Sample Value]
[Listen]	[The Listen address. It can include the host name and port or just the port.]	[[orc13.example.com:8888#or#8888]]
[User]	[The Oracle HTTP Server administration user.]	[[[admin_user]]]
[Group]	[The group for the user.]	[[[admin_group1]]]
[ServerAdmin]	[The administrator's email address.]	[[Webmaster@example.com]]
[ServerName]	[The name of the server for Oracle HTTP Server. If the host does not have a registered DNS name, use the IP address.]	[[orc11.example.com]]
[WebLogicHost]	[The name of the host on which Oracle WebLogic Server is listening for requests.]	[[orc12.example.com]]
[WebLogicPort]	[The port number that Oracle WebLogic Server uses to listen for requests.]	[[9002]]
[WebLogicCluster]	[The name of the host on which an Oracle WebLogic Server cluster is running and its port number.]	[[orc13.example.com:9003]]
[VirtualHost]	[The name of the virtual host. The port number listed should also be listed in the Listen directive.]	[[*.8888]]

Table A-17 [(Cont.)]Move Plan Properties for Oracle HTTP Server

[Property]	[Description]	[Sample Value]
[PlsqlDatabasePassword]	[Specific to the PLSQL module, the name of a secure file containing the password. You must provide a password file, even if you are not changing the configuration.]	[[/scratch/orcl/plsql_passwd.txt]]
[PlsqlDatabaseConnectString]	[Specific to the PLSQL module, the service name of the database.]	[[orcl.example.com:1521:orcl1]]
[PlsqlNLSLanguage]	[Specific to the PLSQL module, the NLS_LANG variable for the database access descriptor (DAD).]	[[America_America.UTF8]]
[ORACONNECTSN]	[Specific to the oradav module, the Oracle database to which to connect.]	[[[db_host:db_port:db_service_name]]]
[ORAUSER]	[Specific to the oradav module, the database user (schema) to use when connecting to the service specified by the ORACONNECTSN property.]	[[db6175_PORTAL]]
[ORACRYPTPASSWORD]	[Specific to the oradav module, the absolute path to the secure file containing the password for oradav. You must provide a password file, even if you are not changing the configuration.]	[[/scratch/oracle/dav_passwd.txt]]
[SSLWallet]	[The location of the SSL wallet, if the wallet is not in the default location.]	[[/scratch/oracle/base_domain/config/fmwconfig/components/OHS/ohs_1/keystores/mywallets]]
[DocumentRoot]	[The directory that stores the main content for the Web site.]	[[/scratch/oracle/base_domain/config/fmwconfig/components/OHS/instances/ohs_1/htdocs]]
[Alias]	[The location of the alias, if it is not in the default location. Note that you change the value within the double quotation marks.]	[[/icons/" /scratch/orcl/icons/"]]
[ScriptAlias]	[The location of the script alias, if it is not in the default location. Note that you change the value within the double quotation marks.]	[[/cgi-bin/" /scratch/oraclcgi-bin/"]]
[WebGateInstallDir]	[The location of the WebGate installation directory, as specified in the webgate.conf file.]	[[/scratch/oracle/oh_home/Oracle_OAMWebGate1/webgate/ohs]]
[primaryOAMServerHost]	[The primary Oracle Access Manager server host.] [Note that the configuration for the secondary Oracle Access Manager server host is updated automatically the first time that WebGate communicates with the primary server.]	[[[primary_oam_server_host].example.com]]
[primaryOAMServerPort]	[The port number for the Oracle Access Manager primary host.]	[[5575]]

[[Table A-18] describes the move plan properties that you can change for Oracle Web Services Manager. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment. Do not edit the value of the configProperty id.]

Table A-18 [J]Move Plan Properties for Oracle Web Services Manager

[Property]	[Description]	[Sample Value]
[[Properties in the bootstrap.configmanager componentType]]	[The bootstrap properties used by the Oracle Web Services Manager agent to connect to the Oracle Web Services Manager Policy Manager.]	[[[]]]
[pm.url]	[The URL of the policy manager. If you are using the same policy manager in the target environment as in the source environment, you do not need to modify this property.]	[[t3://example.com:7003]]
[keystore.path]	[The keystore path. This can be an absolute path or a path relative to the [DOMAIN_HOME]/config/fmwconfig directory.]	[[/scratch/oracle/domains/[domain_name]/config/fmwconfig/default-keystore.jks]]
[truststore.path]	[The truststore path. This can be an absolute path or a path relative to the [DOMAIN_HOME]/config/fmwconfig directory.]	[[/scratch/oracle/domains/[domain_name]/config/fmwconfig/default-keystore.jks]]
[[Properties in the wsm.respository componentType]]	[The Oracle Web Services Manager repository configuration in the move plan if the Policy Manager is installed in the domain. However, if you set the -mdsDataExport parameter of the copyConfig script to false, this component type will not be present in the move plan.] [This component will not be present if wsm-pm is not installed in the domain.]	[[[]]]
[[Properties in the configurations configGroup]]	[The domain configuration. This configGroup is present in the move plan if applications or domains are registered in the source environment or domain configuration documents are created on the source environment.] [There is one configGroup per domain.]	[[[]]]
[domain]	[The domain name of the context for which a configuration document is created in the repository. This property is present in the move plan when you have made changes to the default configuration for a particular context.] [If you do not modify this property, the configurations properties will have same context as on source environment.]	[[domain1]]
[KerberosLoginModule:principal]	[The name of the principal to be used.]	[[HOST/localhost@example.com#]]
[KerberosLoginModule:keytab]	[The path to the keytab to get the secret key for the principal. This can be an absolute path or a path relative to the [DOMAIN_HOME]/config/fmwconfig directory.]	[[./mylocation/krb5.keytab]]
[ConfigManager:pm.url]	[The URL that specifies the location of the policy accessor.]	[[t3://host.example.com:7003]]
[ConfigManager:keystore.path]	[The path to the keystore. This can be an absolute path or a path relative to the [DOMAIN_HOME]/config/fmwconfig directory.]	[[./mylocation/mykeystore.jks#]]

Table A–18 [(Cont.)]Move Plan Properties for Oracle Web Services Manager

[Property]	[Description]	[Sample Value]
[ConfigManager:truststore.path]	[The truststore path relative to the domain configuration directory. This can be an absolute path or a path relative to the [DOMAIN_HOME]/config/fmwconfig directory.]	[[/mylocation/mykeystore.jks#]]
[KeystoreConfig:location]	[The location of the keystore used for message protection. This can be an absolute path or a path relative to the [DOMAIN_HOME]/config/fmwconfig directory.]	[[mycustomlocation/key.tab]]
[[Properties in the policysets configGroup]]	[The configuration of policy sets. This configGroup is present in the move plan if policy sets have been created in the source environment.]	[[[]]]
[attach To]	[The scope of the policy sets. If the attachTo expression scope is DOMAIN or DOMAIN [term], this property is present in the move plan. However, an exception is if it is [DOMAIN(" * ")]. In this case, the property will not be present in the move plan.]	[[domain1]]
[[Properties in the policies configGroup]]	[The configuration of policies. This configGroup is present in the move plan if policies with assertions have been modified in the source environment. There can be multiple configGroups, one for each policy. The ID attribute is the name of the policy for which configGroup is created.]	[[[]]]
[wsdl-uri]	[The actual endpoint URI of the WSDL.]	[[http://[host:port]/sts?wsdl]]
[port-uri]	[The actual endpoint URI of the STS port.]	[[http://host:port/sts-service]]
[sts.auth.service.principal.name]	[The name of the principal that should be used by the service.]	[[HOST/localhost@EXAMPLE.COM]]
[sts.auth.keytab.location]	[The location of the client's keytab file. This property is present when Security Token Service (STS) is configured with Kerberos tokens.]	[[mycustomlocation/key.tab]]
[sts.auth.caller.principal.name]	[Client's principal name.]	[[testuser]]
[service.principal.name]	[The name of the principal that should be used by the service.]	[[HOST/localhost@EXAMPLE.COM]]
[caller.principal.name]	[Client's principal name.]	[[testuser]]
[keytab.location]	[The location of the client's keytab file.]	[[mycustomlocation/key.tab]]

[[Table A–19] describes the move plan properties that you can change for Oracle Data Integrator. Edit all properties, such as host names, port numbers, listen addresses, that have different values in the target environment.]

Table A-19 [J]Move Plan Properties for Oracle Data Integrator

[Property]	[Description]	[Sample Value]
[[Properties in the Master Repository configGroup:]]	[Master repository configuration]	[]
[Url]	[JDBC URL for connecting to the master repository.]	[[jdbc:oracle:thin:@localhost:1522:orc1]]
[Schema name]	[The name of the schema in the target database where the target ODI repository will be created.]	[[odi_master_12c]]
[Schema password file]	[The absolute path of a secure file that contains the password for the schema.]	[[/scratch/oracle/odi_schema_passwd.txt]]
[SUPERVISOR user]	[The name of the ODI user SUPERVISOR.]	[[SUPERVISOR]]
[SUPERVISOR password file]	[The absolute path of a secure file that contains the password for the ODI user SUPERVISOR.]	[[/scratch/oracle/odi_passwd.txt]]
[[Properties in the Physical Data Servers configProperty:]]	[Data servers configuration]	[]
[Schema name]	[The name of the schema for the database data servers or the directory location for file type data servers.]	[[FG_Dir_Schema]]
[Work Schema]	[The name of the Work schema for the database data servers or the directory location for file type data servers.]	[[/tmp/FG_Dir_Schema]]
[Url]	[JDBC URL for connecting to the data server.]	[[jdbc:oracle:thin:@localhost:1521/example.com]]
[User]	[User name for the physical data servers connection.]	[[username]]
[Password File]	[The absolute path of a secure file that contains the password for the user for the physical data servers connection.]	[[/scratch/oracle/rpd_ds_conn_passwd.txt]]
[[Properties in the Agents configProperty:]]	[Agents configuration]	[]
[Host name]	[The Agent host name.]	[[localhost]]
[Host port]	[The Agent host port number.]	[[12311]]
[[Properties in the Work Repositories configProperty:]]	[Work repositories configuration]	[]
[Url]	[JDBC URL for connecting to the work repository.]	[[jdbc:oracle:thin:@localhost:1521/example.com]]
[User]	[User name for connecting to the work repository.]	[[username]]
[Password File]	[The absolute path of a secure file that contains the password for the user for the physical data servers connection.]	[[/scratch/oracle/odi_pds_passwd.txt]]

Oracle Fusion Middleware Command-Line Tools

This appendix summarizes the command-line tools that are available in Oracle Fusion Middleware.

Table B-1 Oracle Fusion Middleware Command-Line Tools

Command	Path	Description
adrci	UNIX:#ORACLE_HOME/bin/adr/adrci.sh Windows:#ORACLE_HOME\bin\adrci.bat	Package incident and problem information into a zip file for transmission to Oracle Support.
config	UNIX:#ORACLE_HOME/oracle_common/common/bin/config.sh Windows:ORACLE_HOME\oracle_common\common\bin\config.cmd	Invoke the Configuration Wizard to create and configure a domain or extend a domain. See: The Installation Guide for the component.
ua	UNIX:#ORACLE_HOME/oracle_common/upgrade/bin#/ua Windows:#ORACLE_HOME\oracle_common\upgrade\ua.bat	Oracle Fusion Middleware Upgrade Assistant. See: <i>Planning an Upgrade of Oracle Fusion Middleware</i>
orapki	UNIX:#ORACLE_HOME/oracle_common/bin/orapki Windows:#ORACLE_HOME\oracle_common\bin\orapki.bat	Manages wallets and certificates. See Appendix G .
wlst	UNIX:#ORACLE_HOME/oracle_common/common/bin/wlst.sh Windows:#ORACLE_HOME\oracle_common\common\bin\wlst.cmd##	(WebLogic Scripting tool) Manages Oracle WebLogic Server and the components in an Oracle WebLogic Server domain. See: Section 2.4, WLST Command Reference for WebLogic Server , and WLST Command Reference for Infrastructure Components

URLs for Components

This appendix provides the URLs needed to access Oracle Fusion Middleware components.

[Table C-1](#) shows the URLs to access components after installation.

The URLs in the table are shown with the default ports. The components in your environment might use different ports. To determine the port numbers, from the WebLogic Domain menu in Fusion Middleware Control, select **Port Usage**.

Table C-1 *URLs for Components*

Component	URL (with Default Port Number)
Oracle Enterprise Manager Fusion Middleware Control	<code>http://host:7001/em</code>
Oracle HTTP Server	<code>http://host:7777</code>
Oracle WebLogic Server Administration Console	<code>http://host:7001/console</code>

Port Numbers

This appendix provides information about Oracle Fusion Middleware port numbers. It contains the following sections:

- [Port Numbers by Component](#)
- [Port Numbers \(Sorted by Number\)](#)

D.1 Port Numbers by Component

This section provides the following information for each Oracle Fusion Middleware component or service that uses a port:

- **Component or Service:** The name of the component and service.
- **Default Port Number:** The first port number Oracle Fusion Middleware attempts to assign to a component. It is usually the lowest number in the allotted port range. If the port is in use, the next available port number, within the allotted range, is assigned.
- **Allotted Port Range:** The set of port numbers Oracle Fusion Middleware attempts to use when assigning a port.

Port numbers for Oracle WebLogic Server servers are assigned sequentially for each server created. For example, the first Administration Server is assigned the port 7001, the second 7002. Managed Servers created during installation and configuration for particular components may have specific default port numbers.

[Table D-1](#) shows the default port number and the port number range for components, sorted alphabetically by component.

Table D-1 Port Numbers Sorted by Component

Component or Service	Default Port Number	Allotted Port Range
Oracle Data Integrator	15000	15500
Oracle HTTP Server non-SSL Listen Port	7777 or 8888	7777-7877, 8888
Oracle HTTP Server SSL Listen Port	4443	4443-4543
Oracle WebLogic Server Listen Port for Administration Server	7001	7001-9000
Oracle WebLogic Server Listen Port for Managed Server	8001	8000 - 8080
Oracle WebLogic Server Node Manager Port	5556	5556

Table D-1 (Cont.) Port Numbers Sorted by Component

Component or Service	Default Port Number	Allotted Port Range
Oracle WebLogic Server SSL Listen Port for Administration Server	7002	7002-9000

D.2 Port Numbers (Sorted by Number)

[Table D-2](#) lists Oracle Fusion Middleware ports numbers and components, sorted in ascending order by port number.

Table D-2 Port Numbers Sorted by Number

Default Port Number	Component or Service
4443	Oracle HTTP Server (SSL)
5556	Oracle WebLogic Server Node Manager Port
7001	Oracle WebLogic Server Listen Port for Administration Server
7002	Oracle WebLogic Server SSL Listen Port for Administration Server
7777	Oracle HTTP Server (non-SSL)
15000	Oracle Data Integrator

Using Oracle Fusion Middleware Accessibility Options

This appendix includes information about using Oracle Fusion Middleware accessibility options.

It includes the following sections:

- [Install and Configure Java Access Bridge \(Windows Only\)](#)
- [Enabling Fusion Middleware Control Accessibility Mode](#)
- [Fusion Middleware Control Keyboard Navigation](#)

E.1 Install and Configure Java Access Bridge (Windows Only)

If you are installing on a Windows computer, you can install and configure Java Access Bridge for Section 508 Accessibility:

1. Download Java Access Bridge from the following URL:
<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136191.html>
2. Install Java Access Bridge.
3. Copy the `access-bridge.jar` and `jaccess-1_4.jar` files from your installation location to the `jre/lib/ext` directory.
4. Copy the `WindowsAccessBridge.dll`, `JavaAccessBridge.dll`, and `JAWTAccessBridge.dll` files from your installation location to the `jre/bin` directory.
5. Copy the `accessibility.properties` file to the `jre/lib` directory.

E.2 Enabling Fusion Middleware Control Accessibility Mode

The following sections provide information on the benefits of running Fusion Middleware Control in accessibility mode, as well as instructions for enabling accessibility mode:

- [Making HTML Pages More Accessible](#)
- [Viewing Text Descriptions of Fusion Middleware Control Charts](#)

E.2.1 Making HTML Pages More Accessible

In Fusion Middleware Control, you can enable screen reader support. Screen reader support improves behavior with a screen reader. This is accomplished by adding

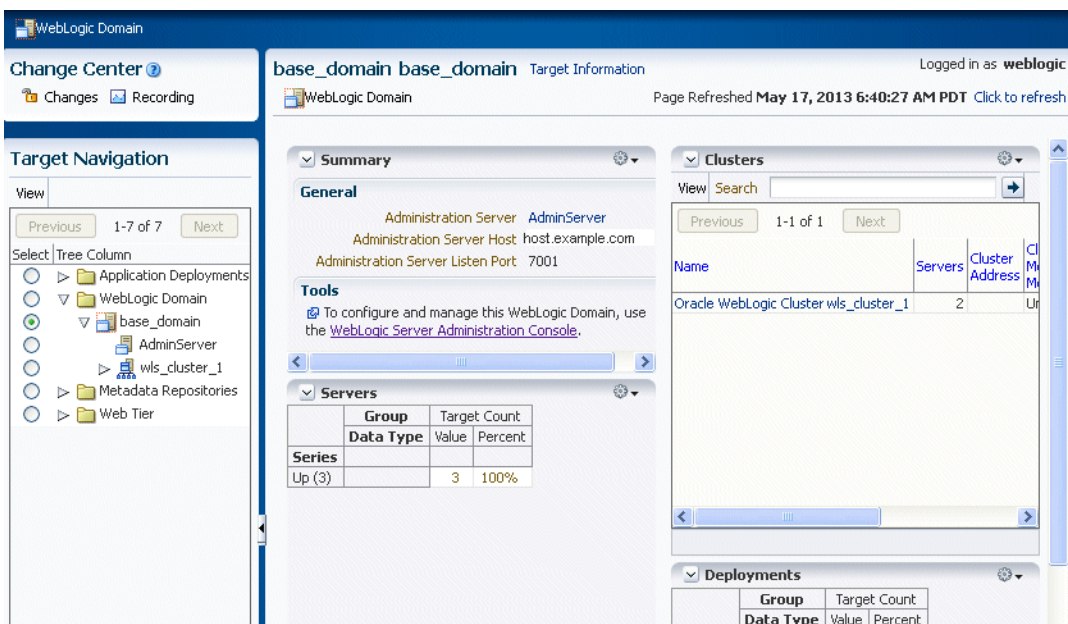
accessibility-specific constructs to the HTML, and by altering some navigation elements on the pages.

To enable screen reader mode in Fusion Middleware Control:

1. Choose the user name at the right top of the page, then **Accessibility**.
The Accessibility Preference page is displayed.
2. Select any of the following options:
 - **I use a screen reader:** Accessibility-specific constructs are added to improve behavior with a screen reader.
 - **I use high contrast settings:** The fonts use a high contrast.
 - **I use large fonts:** The fonts are larger than normal.
 - **Show me the Accessibility Preference dialog when I log in:** When you log in, the Accessibility Preference page is displayed.
3. Click **Apply**.
A confirmation dialog box is displayed.
4. Click **OK**.
5. Click Enterprise Manager at the top of the page to return to the page you last visited.

When you select screen reader support, Fusion Middleware Control renders the Web pages so that they can be read by a screen reader. For example, each node in the navigation tree includes a Select button.

The following figure shows the navigation pane and the Administration Server Performance Summary after enabling screen reader support:



E.2.2 Viewing Text Descriptions of Fusion Middleware Control Charts

Throughout Fusion Middleware Control, charts are used to display performance data. For most users, these charts provide a valuable graphical view of the data that can

reveal trends and help identify minimum and maximum values for performance metrics.

However, charts do not convey information in a manner that can be read by a screen reader. To remedy this problem, you can configure Fusion Middleware Control to provide a complete textual representation of each performance chart. When you enable screen reader mode, Fusion Middleware Control displays the information in tables, instead of charts.

To view a representation of the data in a table, instead of a chart, without enabling screen reader mode, click **Table View** below a chart.

E.3 Fusion Middleware Control Keyboard Navigation

This section describes the keyboard navigation in Fusion Middleware Control.

Much of the keyboard navigation is the same whether or not you use screen reader mode.

Generally, you use the following keys to navigate:

- Tab key: Move to the next control, such as a dynamic target menu, navigation tree, content pane, or tab in a page. Tab traverses the page left to right, top to bottom. Use Shift +Tab to move to the previous control.
- Up and Down Arrow keys: Move to the previous or next item in the navigation tree, menu, or table. Down Arrow also opens a menu.
- Left and Right Arrow keys: Collapse and expand an item in the navigation tree or a submenu.
- Esc: Close a menu.
- Spacebar: Activate a control. For example, in a check box, spacebar toggles the state, checking or unchecking the box. On a link, spacebar navigates to the target of the link.
- Enter: Activate a button.

Table E-1 shows some common tasks and the keyboard navigation used.

Table E-1 Keyboard Navigation for Common Tasks

Task	Navigation
Move to next control, such as navigation tree or menu	Tab
Move to previous control, such as navigation tree or menu	Shift+Tab
Move to navigation pane	Tab until navigation tree has input focus
Move down the navigation tree	Down Arrow
Move up the navigation tree	Up Arrow
Expand a folder	Right Arrow
Collapse a folder	Left Arrow
Open a menu	Down Arrow
Move to the next item in a menu	Down Arrow
Move to the previous item in a menu	Up Arrow
Select a menu item	Enter

Table E-1 (Cont.) Keyboard Navigation for Common Tasks

Task	Navigation
Open a submenu	Right Arrow
Close a submenu	Left Arrow
Move out of a menu	Esc
Activate a button	Enter
Open a tab in a content pane	Tab to the content pane, Tab to the tab to get input focus, then Enter to select the tab
Select an item, such as Message type in Log Messages screen	Spacebar
Select a row in a table	Tab to the header of the table, then Down Arrow to move to a row
Select a cell in a table	Tab to the header of the table, then Tab until you reach the cell you want to select, then Enter

Table E-2 shows the keyboard navigation for the Topology Viewer. The navigation from one node to another is based on the geometry of the topology.

Table E-2 Keyboard Navigation for Topology Viewer

Task	Navigation
Navigate into the topology	Tab, until you have reached a node.
Navigate nodes based on geometry	Arrow keys
In a top-to-bottom orientation, navigate to a destination link	Ctrl+Shift+Down Arrow
In a top-to-bottom orientation, navigate to a source link	Ctrl+Shift+Up Arrow
In a left-to-right orientation, navigate to a destination link	Ctrl+Shift+Right Arrow
In a left-to-right orientation, navigate to a source link	Ctrl+Shift+Left Arrow
In a top-down orientation, when on a link, navigate to other links. The focus moves to another link based on the geometry.	Right Arrow or Left Arrow
In a left-to-right orientation, when on a link, navigate to other links. The focus moves to another link based on the geometry.	Up Arrow or Down Arrow
Move into or out of a group node	Shift+Arrow
Simulate a mouse click on the node. This can bring up a popup or it can navigate to another page.	Enter
Simulate a mouse over. Typically, this brings up a popup.	Shift+Enter
Simulate a right-mouse click. Typically, this brings up a context menu.	m
Expand or contract a node subtree	e

Table E-2 (Cont.) Keyboard Navigation for Topology Viewer

Task	Navigation
Expand or contract a group. Note that if you use Shift+Arrow to move into a group, the group automatically expands.	g
Pan up or down, left or right	Ctrl+Arrow keys
Zoom in	Ctrl+Alt+Plus Key(+)
Zoom out	Ctrl+Alt+Minus Key(-)
Move out of a menu.	Esc

Viewing Release Numbers

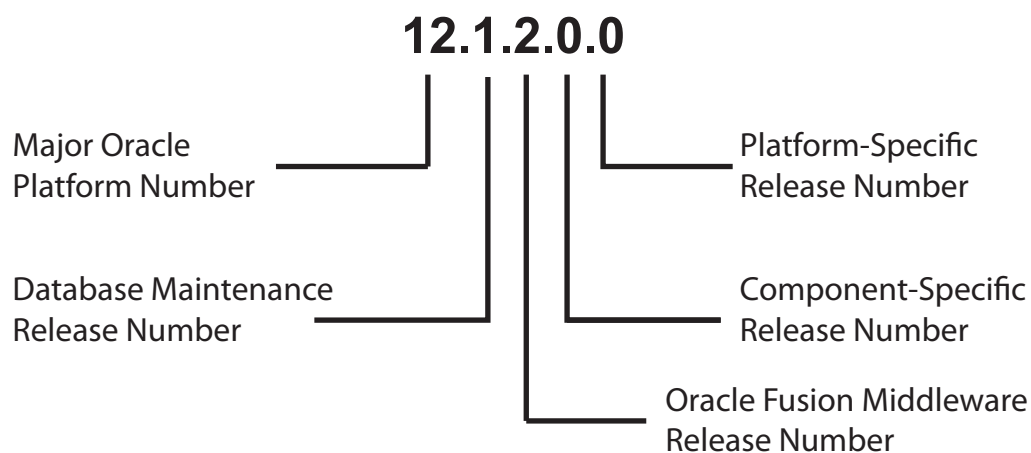
This appendix describes how to view Oracle Fusion Middleware release numbers. It appendix contains the following sections:

- [Release Number Format](#)
- [Viewing the Software Inventory and Release Numbers](#)

F.1 Release Number Format

To understand the release level nomenclature used by Oracle, examine the example of an Oracle Fusion Middleware release number shown in [Figure F-1](#).

Figure F-1 Example of an Oracle Fusion Middleware Release Number



In [Figure F-1](#), each digit is labeled:

- Major Oracle platform number
This is the most general identifier. It represents a major new edition (or version) of an application, such as Oracle database server or Oracle Fusion Middleware, and indicates that the release contains significant new functionality.
- Database maintenance release number
This digit represents a maintenance release level. Some new features may also be included.
- Oracle Fusion Middleware release number
This digit reflects the release level of Oracle Fusion Middleware.

- Component-specific release number
This digit identifies a release level specific to a component. Different components can have different numbers in this position depending upon, for example, component patch sets or interim releases.
- Platform-specific release number
This digit identifies a platform-specific release.

F.2 Viewing the Software Inventory and Release Numbers

The following sections describe how to obtain the release numbers of Oracle Fusion Middleware:

- [Viewing Oracle Fusion Middleware Installation Release Numbers](#)
- [Viewing Component Release Numbers](#)
- [Viewing Metadata Repository Release Numbers](#)

F.2.1 Viewing Oracle Fusion Middleware Installation Release Numbers

All Oracle Fusion Middleware installations have a release number. This number is updated when you apply a patch set release or upgrade the installation.

You can view the release number of an Oracle Fusion Middleware installation using Opatch. Run the following command:

```
(UNIX) #ORACLE_HOME/OPatch/opatch#lsinventory
(Windows) #ORACLE_HOME\OPatch\opatch#lsinventory
```

For example, on UNIX:

```
#!/opatch#lsinventory
Oracle#Interim#Patch#Installer#version#13.1.0.0.0
Copyright#(c)#2013,#Oracle#Corporation.##All#rights#reserved.
#
#
Oracle#Home#####:/scratch/oracle1/Oracle/Middleware/Oracle_Home
Central#Inventory#:/scratch/oracle1/oraInventory
###from#####:/scratch/oracle1/Oracle/Middleware/Oracle_Home/oraInst.loc
OPatch#version####:#13.1.0.0.0
OUI#version#####:#13.1.0.0.0
Log#file#location#:/scratch/oracle1/Oracle/Middleware/Oracle_
Home/cfgtoollogs/opatch/opatch2013-06-07_08-24-23AM_1.log
#
#
OPatch#detects#the#Middleware#Home#as#" /scratch/oracle1/Oracle/Middleware/Oracle_
Home"

Lsinventory#Output#file#location#:/scratch/oracle1/Oracle/Middleware/Oracle_
Home/cfgtoollogs/opatch/lsinv/lsinventory2013-06-07_08-24-23AM.txt
#
-----
.
.
.#
```

F.2.2 Viewing Oracle WebLogic Server Release Numbers

You can use the following command to view the release number of Oracle WebLogic Server:

```
(UNIX) #cat#$ORACLE_HOME/wlserver/.product.properties#|#grep#WLS_PRODUCT_VERSION
(Windows) #type#%ORACLE_HOME%\wlserver\product.properties#|#findstr#WLS_PRODUCT_VERSION
```

For example, on UNIX:

```
cat#$ORACLE_HOME/wlserver/.product.properties#|#grep#WLS_PRODUCT_VERSION
WLS_PRODUCT_VERSION=10.3.6.0
```

F.2.3 Viewing Component Release Numbers

All Oracle Fusion Middleware components have a release number and many contain services that have release numbers. These numbers *may* be updated when you apply a patch set release or upgrade the installation.

You can view the release number of components and their services by using the following commands:

- On UNIX:

```
cd#ORACLE_HOME/inventory
ls#-d#Components*/**/*
```

- On Windows:

```
cd#ORACLE_HOME/inventory/Componentsn
dir#/S#/A:D
```

F.2.4 Viewing Metadata Repository Release Numbers

If you are using an Oracle Database instance for your metadata repository, you can view the release number of the database using SQL*Plus as follows (you can be connected to the database as any user to issue these commands):

```
SQL>#COL#PRODUCT#FORMAT#A40
SQL>#COL#VERSION#FORMAT#A15
SQL>#COL#STATUS#FORMAT#A15
SQL>#SELECT#*#FROM#PRODUCT_COMPONENT_VERSION;

PRODUCT#####VERSION#####STATUS
-----#-----#-----
NLSRTL#####11.2.0.1.0#####Production
Oracle#Database#11g#Enterprise#Edition#####11.2.0.1.0#####Production
PL/SQL#####11.2.0.1.0#####Production
TNS#for#Linux:#####11.2.0.1.0#####Production
```

F.2.5 Viewing Schema Release Numbers

If you are using an Oracle Database instance for your metadata repository, you can view the release number of the schema using SQL*Plus, as follows:

```
SQL>#COL#COMP_ID#FORMAT#A20
SQL>#COL#COMP_NAME#A30
SQL>#COL#VERSION#FORMAT#A20
SQL>#SELECT#COMP_ID, #COMP_NAME, #VERSION#FROM#SCHEMA_VERSION_REGISTRY;
#
COMP_NAME#####VERSION
```

```
-----#-----#-----  
Audit#Service#####12.1.2.0.0  
Audit#Servicei#Append#####12.1.2.0.0#  
Audit#Service#Viewer#####12.1.2.0.0  
Metadata#Services#####12.1.2.0.0#  
#.  
#.  
#.
```

Oracle Application Server 10g provided the `orapki` utility, a command-line tool to manage certificate revocation lists (CRLs), create and manage Oracle wallets, and create signed certificates for testing purposes. It also provided the SSL Configuration Tool.

Oracle Fusion Middleware 12c (12.1.2) provides both command-line and graphical user interfaces to configure SSL. The Oracle WebLogic Scripting Tool (WLST) and Oracle Enterprise Manager Fusion Middleware Control enable you to manage KSS- and JKS-based keystores, wallets, and certificates.

Use this appendix to learn about how to transition from pre-12c tools like `orapki` to the certificate, wallet management, and SSL configuration tools provided in 12c (12.1.2). The appendix contains these topics:

- [Using the orapki Utility for Certificate Validation and CRL Management](#)
- [Equivalent Features for orapki](#)
- [Features of the SSL Configuration Tool](#)

See Also: :

- *Oracle Advanced Security Administrator's Guide* for details of `orapki` usage:
http://docs.oracle.com/cd/E11882_01/network.112/e10746/toc.htm
- [Chapter 6](#) for details about keystore and wallet management in Oracle Fusion Middleware.

Note: The `orapki` utility is located in the binary directory of Oracle Common home, that is, `$ORACLE_HOME/oracle_common/bin`.

G.1 Using the `orapki` Utility for Certificate Validation and CRL Management

This section contains these topics:

- [orapki Overview](#)
- [Displaying orapki Help](#)
- [Creating Signed Certificates for Testing Purposes](#)
- [Managing Oracle Wallets with the orapki Utility](#)

- [Managing Certificate Revocation Lists \(CRLs\) with orapki Utility](#)
- [orapki Utility Commands Summary](#)

G.1.1 orapki Overview

The `orapki` utility is provided to manage public key infrastructure (PKI) elements, such as wallets and certificate revocation lists, on the command line so the tasks it performs can be incorporated into scripts. This enables you to automate many of the routine tasks of maintaining a PKI.

This command-line utility can be used to perform the following tasks:

- Creating signed certificates for testing purposes
- Managing Oracle wallets:
 - Creating and displaying Oracle wallets
 - Adding and removing certificate requests
 - Adding and removing certificates
 - Adding and removing trusted certificates
- Managing certificate revocation lists (CRLs):
 - Renaming CRLs with a hash value for certificate validation

`orapki` allows you to import certificates in both DER and PEM formats.

G.1.1.1 orapki Syntax

The basic syntax of the `orapki` command-line utility is as follows:

```
orapki#module#command#-parameter#value
```

In the preceding command, *module* can be `wallet` (Oracle wallet), `crl` (certificate revocation list), or `cert` (PKI digital certificate). The available commands depend on the module you are using. For example, if you are working with a `wallet`, then you can add a certificate or a key to the wallet with the `add` command. The following example adds the user certificate located at `/private/lhale/cert.txt` to the wallet located at `ORACLE_HOME/wallet/ewallet.p12`:

```
orapki#wallet#add#-wallet#ORACLE_HOME/wallet/ewallet.p12  
-user_cert#-cert#/private/lhale/cert.txt
```

G.1.1.2 Environment Setup for orapki

When running `orapki` in the context of Web Tier or Classic installations, set `ORACLE_HOME` to point to the product installation location.

G.1.2 Displaying orapki Help

You can display all the `orapki` commands that are available for a specific mode by entering the following at the command line:

```
orapki#mode#help
```

For example, to display all available commands for managing certificate revocation lists (CRLs), enter the following at the command line:

```
orapki#crl#help
```

Note: Using the `-summary`, `-complete`, or `-wallet` command options is always optional. A command will still run if these command options are not specified.

G.1.3 Creating Signed Certificates for Testing Purposes

This command-line utility provides a convenient, lightweight way to create signed certificates for testing purposes. The following syntax can be used to create signed certificates and to view certificates:

To create a signed certificate for testing purposes:

```
orapki#cert#create#[-wallet#wallet_location]#-request
#certificate_request_location
-cert#certificate_location#-validity#number_of_days#[-summary]
```

This command creates a signed certificate from the certificate request. The `-wallet` parameter specifies the wallet containing the user certificate and private key that will be used to sign the certificate request. The `-validity` parameter specifies the number of days, starting from the current date, that this certificate will be valid. Specifying a certificate and certificate request is mandatory for this command.

To view a certificate:

```
orapki#cert#display#-cert#certificate_location#[-summary#|#-complete]
```

This command enables you to view a test certificate that you have created with `orapki`. You can choose either `-summary` or `-complete`, which determines how much detail the command will display. If you choose `-summary`, the command will display the certificate and its expiration date. If you choose `-complete`, it will display additional certificate information, including the serial number and public key.

G.1.4 Managing Oracle Wallets with the orapki Utility

The following sections describe the syntax used to create and manage Oracle wallets with the `orapki` command-line utility. You can use these `orapki` utility `wallet` module commands in scripts to automate the wallet creation process.

- [Creating and Viewing Oracle Wallets with orapki](#)
- [Adding Certificates and Certificate Requests to Oracle Wallets with orapki](#)
- [Exporting Certificates and Certificate Requests from Oracle Wallets with orapki](#)

Note: The `-wallet` parameter is mandatory for all `wallet` module commands.

See Also: For examples of how to create either a password-protected wallet or an auto-login wallet, see Doc ID 1226654.1, "How To Create a Wallet via ORAPKI in FMW 11g" on the OTN Knowledge Base.

G.1.4.1 Creating and Viewing Oracle Wallets with orapki

To create an Oracle wallet:

```
orapki#wallet#create#[-wallet#wallet_location]
```

This command will prompt you to enter and re-enter a wallet password. It creates a wallet in the location specified for `-wallet`.

To create an Oracle wallet with auto-login enabled:

```
orapki#wallet#create#-wallet#wallet_location#-auto_login
```

This command creates a wallet with auto-login enabled, or it can also be used to enable auto-login on an existing wallet. If the `wallet_location` already contains a wallet, then auto-login will be enabled for it. To disable the auto-login feature, delete `cwallet.sso`.

Note: For wallets with the auto-login feature enabled, you are prompted for a password only for operations that modify the wallet, such as `add`.

To view an Oracle wallet:

```
orapki#wallet#display#-wallet#wallet_location
```

This command displays the certificate requests, user certificates, and trusted certificates contained in the wallet.

G.1.4.2 Adding Certificates and Certificate Requests to Oracle Wallets with orapki

To add a certificate request to an Oracle wallet:

```
orapki#wallet#add#-wallet#wallet_location#-dn#user_dn#-keysize#512|1024|2048|4096
```

This command adds a certificate request to a wallet for the user with the specified distinguished name (`user_dn`). The request also specifies the requested certificate's key size (512, 1024, or 2048 bits). To sign the request, export it with the `export` option. See [Section G.1.4.3, "Exporting Certificates and Certificate Requests from Oracle Wallets with orapki."](#)

To add a trusted certificate to an Oracle wallet:

```
orapki#wallet#add#-wallet#wallet_location#-trusted_cert#-cert_certificate_location
```

This command adds a trusted certificate, at the specified location (`-cert certificate_location`), to a wallet. You must add all trusted certificates in the certificate chain of a user certificate before adding a user certificate, or the command to add the user certificate will fail.

To add a root certificate to an Oracle wallet:

```
orapki#wallet#add#-wallet#wallet_location#-dn_certificate_dn#-keysize#512|1024|2048#-self_signed#-validity#number_of_days
```

This command creates a new self-signed (root) certificate and adds it to the wallet. The `-validity` parameter (mandatory) specifies the number of days, starting from the current date, that this certificate will be valid. You can specify a key size for this root certificate (`-keysize`) of 512, 1024, 2048, or 4096 bits.

To add a user certificate to an Oracle wallet:

```
orapki#wallet#add#-wallet#wallet_location#-user_cert#-cert#certificate_location
```

This command adds the user certificate at the location specified with the `-cert` parameter to the Oracle wallet at the `wallet_location`. Before you add a user certificate to a wallet, you must add all the trusted certificates that make up the certificate chain. If all trusted certificates are not installed in the wallet before you add the user certificate, then adding the user certificate will fail.

G.1.4.3 Exporting Certificates and Certificate Requests from Oracle Wallets with orapki

To export a certificate from an Oracle wallet:

```
orapki#wallet#export#-wallet#wallet_location#-dn
certificate_dn#-cert#certificate_filename
```

This command exports a certificate with the subject's distinguished name (`-dn`) from a wallet to a file that is specified by `-cert`.

To export a certificate request from an Oracle wallet:

```
orapki#wallet#export#-wallet#wallet_location#-dn
certificate_request_dn#-request#certificate_request_filename
```

This command exports a certificate request with the subject's distinguished name (`-dn`) from a wallet to a file that is specified by `-request`.

G.1.5 Managing Certificate Revocation Lists (CRLs) with orapki Utility

CRLs must be managed with `orapki`. This utility creates a hashed value of the CRL issuer's name to identify the CRLs location in your system. If you do not use `orapki`, your Oracle server cannot locate CRLs to validate PKI digital certificates. The following sections describe CRLs, how you use them, and how to use `orapki` to manage them:

- [Section G.1.5.1, "About Certificate Validation with Certificate Revocation Lists"](#)
- [Section G.1.5.2, "Certificate Revocation List Management"](#)

See Also: "Certificate Revocation List Management" in the *Oracle Advanced Security Administrator's Guide* for details about managing CRLs with `orapki`:

http://docs.oracle.com/cd/E11882_01/network.112/e10746/asossl.htm

G.1.5.1 About Certificate Validation with Certificate Revocation Lists

The process of determining whether a given certificate can be used in a given context is referred to as certificate validation. Certificate validation includes determining that:

- A trusted certificate authority (CA) has digitally signed the certificate.
- The certificate's digital signature corresponds to the independently-calculated hash value of the certificate itself and the certificate signer's (CA's) public key.
- The certificate has not expired.
- The certificate has not been revoked.

The SSL network layer automatically performs the first three validation checks, but you must configure certificate revocation list (CRL) checking to ensure that certificates have not been revoked. CRLs are signed data structures that contain a list of revoked

certificates. They are usually issued and signed by the same entity who issued the original certificate.

G.1.5.1.1 What CRLs Should You Use? You should have CRLs for all of the trust points that you honor. The trust points are the trusted certificates from a third-party identity that is qualified with a level of trust. Typically, the certificate authorities you trust are called trust points.

G.1.5.1.2 How CRL Checking Works Certificate revocation status is checked against CRLs which are located in file system directories, or downloaded from the location specified in the CRL Distribution Point (CRL DP) extension on the certificate. If you store your CRLs on the local file system or in the directory, then you must update them regularly. If you use CRL DPs then CRLs are downloaded when the corresponding certificates are first used.

The server searches for CRLs in the following locations in the order listed. When the system finds a CRL that matches the certificate CA's DN, it stops searching.

1. Local file system

The system checks the `sqlnet.ora` file for the `SSL_CRL_FILE` parameter first, followed by the `SSL_CRL_PATH` parameter. If these two parameters are not specified, then the system checks the wallet location for any CRLs.

Note: if you store CRLs on your local file system, then you must use the `orapki` utility to periodically update them. See [Section G.1.5.2.1, "Renaming CRLs with a Hash Value for Certificate Validation."](#)

2. CRL DP

If the CA specifies a location in the CRL DP X.509, version 3, certificate extension when the certificate is issued, then the appropriate CRL that contains revocation information for that certificate is downloaded. Currently, Oracle Advanced Security supports downloading CRLs over HTTP and LDAP.

Notes:

- For performance reasons, only user certificates are checked.
 - Oracle recommends that you store CRLs in the directory rather than the local file system.
-
-

G.1.5.2 Certificate Revocation List Management

Before you can enable certificate revocation status checking, you must ensure that the CRLs you receive from the CAs you use are in a form (renamed with a hash value) or in a location (uploaded to the directory) in which your system can use them. Oracle Advanced Security provides a command-line utility, `orapki`, that you can use to perform the following task:

- [Renaming CRLs with a Hash Value for Certificate Validation](#)

Note: CRLs must be updated at regular intervals (before they expire) for successful validation. You can automate this task by using `orapki` commands in a script.

See Also: Command-Line Tools Overview in the *Oracle Fusion Middleware Reference for Oracle Identity Management* for information about LDAP command-line tools and their syntax.

G.1.5.2.1 Renaming CRLs with a Hash Value for Certificate Validation When the system validates a certificate, it must locate the CRL issued by the CA who created the certificate. The system locates the appropriate CRL by matching the issuer name in the certificate with the issuer name in the CRL.

When you specify a CRL storage location for the **Certificate Revocation Lists Path** field in Oracle Net Manager (sets the `SSL_CRL_PATH` parameter in the `sqlnet.ora` file), use the `orapki` utility to rename CRLs with a hash value that represents the issuer's name. Creating the hash value enables the server to load the CRLs.

On UNIX systems, `orapki` creates a symbolic link to the CRL. On Windows systems, it creates a copy of the CRL file. In either case, the symbolic link or the copy created by `orapki` are named with a hash value of the issuer's name. Then when the system validates a certificate, the same hash function is used to calculate the link (or copy) name so the appropriate CRL can be loaded.

Depending on your operating system, enter one of the following commands to rename CRLs stored in the file system.

To rename CRLs stored in UNIX file systems:

```
orapki#crl#hash#-crl#crl_filename#[-wallet#wallet_location]
-symlink#crl_directory#[-summary]
```

To rename CRLs stored in Windows file systems:

```
orapki#crl#hash#-crl#crl_filename
[-wallet#wallet_location]#-copy#crl_directory#[-summary]
```

In the preceding commands, `crl_filename` is the name of the CRL file, `wallet_location` is the location of a wallet that contains the certificate of the CA that issued the CRL, and `crl_directory` is the directory in which the CRL is located.

Using `-wallet` and `-summary` are optional. Specifying `-wallet` causes the tool to verify the validity of the CRL against the CA's certificate prior to renaming the CRL. Specifying the `-summary` option causes the tool to display the CRL issuer's name.

G.1.6 orapki Utility Commands Summary

This section lists and describes the following `orapki` commands:

- [orapki cert create](#)
- [orapki cert display](#)
- [orapki crl create](#)
- [orapki crl hash](#)
- [orapki crl revoke](#)
- [orapki crl status](#)
- [orapki crl verify](#)
- [orapki wallet add](#)
- [orapki wallet change_pwd](#)
- [orapki wallet create](#)

- [orapki wallet display](#)
- [orapki wallet export](#)
- [orapki wallet export_trust_chain](#)

G.1.6.1 orapki cert create

The following sections describe this command.

G.1.6.1.1 Purpose Use this command to create a signed certificate for testing purposes.

G.1.6.1.2 Syntax `orapki#cert#create#[-wallet#wallet_location]
-request#certificate_request_location
-cert#certificate_location#-validity#number_of_days#[-summary]`

- The `-wallet` parameter specifies the wallet containing the user certificate and private key that will be used to sign the certificate request.
- The `-request` parameter (mandatory) specifies the location of the certificate request for the certificate you are creating.
- The `-cert` parameter (mandatory) specifies the directory location in which the tool places the new signed certificate.
- The `-validity` parameter (mandatory) specifies the number of days, starting from the current date, that this certificate will be valid.

G.1.6.2 orapki cert display

The following sections describe this command.

G.1.6.2.1 Purpose Use this command to display details of a specific certificate.

G.1.6.2.2 Syntax `orapki#cert#display#-cert#certificate_location#
[-summary|-complete]`

- The `-cert` parameter specifies the location of the certificate you want to display.
- You can use either the `-summary` or the `-complete` parameter to display the following information:
 - `-summary` displays the certificate and its expiration date
 - `-complete` displays additional certificate information, including the serial number and public key

G.1.6.3 orapki crl create

The following sections describe this command.

G.1.6.3.1 Purpose Use this command to create a CRL.

G.1.6.3.2 Syntax `orapki#crl#create#[-crl#[url/filename]]#
[-wallet#[cwallet]]#
[-nextupdate#[days]]#
[-pwd#pwd]`

- `-crl` is the location where the CRL will be created (for example `./nzcr1.txt`)
- `-wallet` is the `cawallet`, which contains self-signed certificate and corresponding private key

- `-nextupdate` is the number of days until the next update
- `-pwd` is the password of `cawallet`

G.1.6.4 orapki crl hash

The following sections describe this command.

G.1.6.4.1 Purpose Use this command to generate a hash value of the certificate revocation list (CRL) issuer to identify the location of the CRL in your file system for certificate validation.

G.1.6.4.2 Syntax `orapki#crl#hash#-crl#crl_filename|URL#`
`#[-wallet#wallet_location]#[-symlink|-copy]#crl_directory#[-summary]`

- The `-crl` parameter specifies the filename that contains the CRL or the URL in which it can be found.
- The `-wallet` parameter (optional) specifies the location of the wallet that contains the certificate of the certificate authority (CA) who issued the CRL. Using it causes the tool to verify the validity of the CRL against the CA's certificate prior to uploading it to the directory.
- Depending on your operating system, use either the `-symlink` or the `-copy` parameter:
 - On UNIX: Use `-symlink` to create a symbolic link to the CRL at the `crl_directory` location
 - On Windows: Use `-copy` to create a copy of the CRL at the `crl_directory` location
- The `-summary` parameter (optional) causes the tool to display the CRL issuer's name.

G.1.6.5 orapki crl revoke

The following sections describe this command.

G.1.6.5.1 Purpose Use this command to revoke a certificate.

G.1.6.5.2 Syntax `orapki#crl#revoke#[-crl#[url|filename]]#`
`[-wallet#cawallet]]#`
`[-cert#revokecert]]#`
`[-pwd#pwd]`

where:

- `-crl` specifies the CRL as either a URL or a filename
- `-wallet` is the `cawallet`, which contains self-signed certificate and corresponding private key
- `-cert`: certificate to be revoked
- `-pwd` is the password of `cawallet`.

G.1.6.6 orapki crl status

The following sections describe this command.

G.1.6.6.1 Purpose Use this command to check if a certificate is revoked in a CRL.

G.1.6.6.2 Syntax `orapki#crl#status#[-crl#[url|filename]]#
#[-cert#[cert]]`

- `-crl` specifies the CRL as either a URL or a filename
- `-cert` is the CA's certificate

G.1.6.7 orapki crl verify

The following sections describe this command.

G.1.6.7.1 Purpose Use this command to verify a CRL signature.

G.1.6.7.2 Syntax `orapki#crl#verify#[-crl#[url|filename]]#
[-cert#[cacert]]`

where:

- `-crl` specifies the CRL as either a URL or a filename
- `-cert` specifies the certificate to be checked

G.1.6.8 orapki wallet add

The following sections describe this command.

G.1.6.8.1 Purpose Use this command to add certificate requests and certificates to an Oracle wallet.

G.1.6.8.2 Syntax To add certificate requests:

`orapki#wallet#add#-wallet#wallet_location#-dn#user_dn#-keysize#512|1024|2048`

- The `-wallet` parameter specifies the location of the wallet to which you want to add a certificate request.
- The `-dn` parameter specifies the distinguished name of the certificate owner.
- The `-keysize` parameter specifies the key size for the certificate.
- To sign the request, export it with the export option. See [Section G.1.6.12, "orapki wallet export"](#).

To add trusted certificates:

`orapki#wallet#add#-wallet#wallet_location#-trusted_cert#-cert#certificate_location`

- The `-trusted_cert` parameter causes the tool to add the trusted certificate, at the location specified with `-cert`, to the wallet.

To add root certificates:

`orapki#wallet#add#-wallet#wallet_location#-dn#
certificate_dn#-keysize#512|1024|2048#-self_signed#
-valid_from#[mm/dd/yyyy]#-valid_until#[mm/dd/yyyy]
-validity#number_of_days`

- The `-self_signed` parameter causes the tool to create a root certificate.
- The `-validity` parameter can be used to specify the number of days, starting from the current date, that this root certificate will be valid.

- The `-valid_from` and `valid_until` parameters can be used to specify an exact date range for which this root certificate will be valid. You may specify validity in this way instead of `-validity_number_of_days`.

To add user certificates:

```
orapki#wallet#add#-wallet#wallet_location#-user_cert#-cert#certificate_location
```

- The `-user_cert` parameter causes the tool to add the user certificate at the location specified with the `-cert` parameter to the wallet. Before you add a user certificate to a wallet, you must add all the trusted certificates that make up the certificate chain. If all trusted certificates are not installed in the wallet before you add the user certificate, then adding the user certificate will fail.

To add a subject key identifier extension to a certificate request:

```
orapki#wallet#add#-wallet#wallet_location#-dn#user_dn#-keysize#512|1024|2048#-addext_ski
```

To add a Version 3 self-signed certificate to a wallet:

```
orapki#wallet#add#-wallet#wallet_location#-dn#certificate_dn#-keysize#512|1024|2048#-self_signed#-validity#number_of_days#-addext_ski
```

G.1.6.9 orapki wallet change_pwd

The following sections describe this command.

G.1.6.9.1 Purpose Use this command to change the password for an Oracle wallet.

G.1.6.9.2 Syntax `orapki#wallet#change_pwd#[-wallet#[wallet_location]]#[-oldpwd#oldpassword]#[-newpwd#newpassword]`

- The `-wallet` parameter specifies the location of the wallet whose password you want to change.
- The `-oldpwd` parameter specifies the existing wallet password.
- The `-newpwd` parameter specifies the new wallet password.

G.1.6.10 orapki wallet create

The following sections describe this command.

G.1.6.10.1 Purpose Use this command to create an Oracle wallet or to set auto-login on for an Oracle wallet.

G.1.6.10.2 Syntax `orapki#wallet#create#-wallet#wallet_location#[-auto_login]`

- The `-wallet` parameter specifies a location for the new wallet or the location of the wallet for which you want to turn on auto-login.
- The `-auto_login` parameter creates an auto-login wallet, or it turns on automatic login for the wallet specified with the `-wallet` option.

G.1.6.11 orapki wallet display

The following sections describe this command.

G.1.6.11.1 Purpose Use this command to view the certificate requests, user certificates, and trusted certificates in an Oracle wallet.

G.1.6.11.2 Syntax `orapki#wallet#display#-wallet#wallet_location`

- The `-wallet` parameter specifies a location for the wallet you want to open if it is not located in the current working directory.

G.1.6.12 orapki wallet export

The following sections describe this command.

G.1.6.12.1 Purpose Use this command to export certificate requests and certificates from an Oracle wallet.

G.1.6.12.2 Syntax `orapki#wallet#export#-wallet#wallet_location#-dn certificate_dn#-cert#certificate_filename`

- The `-wallet` parameter specifies the directory where the wallet, from which you want to export the certificate, is located.
- The `-dn` parameter specifies the distinguished name of the certificate.
- The `-cert` parameter specifies the path and filename of the file that contains the exported certificate.

To export a certificate request from an Oracle wallet:

```
orapki#wallet#export#-wallet#wallet_location#-dn
certificate_request_dn#-request#certificate_request_filename
```

- The `-request` parameter specifies the path and filename of the file that contains the exported certificate request.

G.1.6.13 orapki wallet export_trust_chain

The following sections describe this command.

G.1.6.13.1 Purpose Use this command to export a chain of trust (certificate chain) for a user.

G.1.6.13.2 Syntax

```
orapki#wallet#export_trust_chain#[-wallet#[wallet]]#
[-certchain#[filename]]#
[-dn#[user_cert_dn]]#
[-pwd#pwd]
```

- The `-wallet` parameter specifies the location of the wallet from which you want to export the certificate chain.
- The `-certchain` parameter specifies the name of the file to contain the exported certificate chain.
- The `-dn` parameter specifies the distinguished name of the entry to be exported.
- The `-pwd` specifies the wallet password.

G.2 Equivalent Features for orapki

Table G–1 shows the features provided by the `orapki` utility for Oracle wallets and CRLs, and the equivalent commands and options in 12c (12.1.2).

Table G–1 Mapping for orapki Features for Wallets and CRLs

orapki Feature	How Implemented in 11gR1	Notes
Creating a standard PKCS#12 wallet	WLST createWallet()	To manage a password-protected and auto-login wallet, provide a non-empty password value. To manage just an auto-login wallet, provide an empty password value (that is, "")
Creating a PKCS#11 wallet	Not supported	Use orapki command-line tool
Uploading a wallet to an LDAP Directory	Not supported	Use orapki command-line tool
Downloading a wallet from an LDAP directory	Not supported	Use orapki command-line tool
Deleting a wallet	WLST deleteWallet()	
Changing the wallet password	WLST changeWalletPassword()	For obvious reasons, password can only be changed for a password-protected wallet
Enabling auto-login		Auto-login wallet is automatically created with every password-protected wallet.
Enabling auto-login wallet that works only on local machine	Not supported	Use orapki command line tool
Create, revoke, hash, verify, upload, list, display, delete CRLs	Not supported	Use orapki command line tool

Table G–2 shows the features provided by the orapki utility for certificates, and the equivalent commands or options in 12c (12.1.2).

Table G–2 Mapping for orapki Features for Certificates

orapki Feature	How Implemented in WLST in 11gR1	Notes
Adding a certificate request	addCertificateRequest()	
Adding a self-signed certificate	addSelfSignedCertificate()	
Listing all entries in a wallet	listWalletObjects()	Provide a valid value of type ("CertificateRequest", "Certificate" or "TrustedCertificate")
Importing a user certificate	importWalletObject()	Enter type as "Certificate"
Importing a trusted certificate	importWalletObject()	Enter type as "TrustedCertificate"
Removing a certificate request	removeWalletObject()	Enter type as "CertificateRequest"
Removing a user certificate	removeWalletObject()	Enter type as "Certificate"
Removing a trusted certificate	removeWalletObject()	Enter type as "TrustedCertificate"
Removing all trusted certificates	removeWalletObject()	Enter type as "TrustedAll"
Exporting a user certificate	exportKeyStoreObject()	Enter type as "Certificate"
Exporting a certificate request	exportWalletObject()	Enter type as "CertificateRequest"
Exporting a trusted certificate	exportWalletObject()	Enter type as "TrustedCertificate"
Exporting a certificate chain	exportWalletObject()	Enter type as "CertificateChain"
Importing a PKCS#7 certificate chain into the wallet	importWalletObject()	Enter type as "TrustedChain"

G.3 Features of the SSL Configuration Tool

Table G–3 shows the features provided by the SSL Configuration Tool.

Table G-3 Equivalent Features for the SSL Configuration Tool

Configuration Feature	SSL Configuration in 12c (12.1.2)
Support for wallet management	Supports management of Oracle Wallets, JKS Keystores, and KSS Keystores, in addition to SSL configuration
Standalone type supported for SSL	Oracle HTTP Server supported for standalone SSL configuration
Types of interface	Provides both command line interface (WLST) and graphical interface (Fusion Middleware Control)
Configuration file required?	Configuration file is optional in the WLST command. If not provided, default values are used for SSL attributes.
Supported tiers	Supports SSL configuration for both Web tier and data tier.
Local or remote management	Allows remote management of components.

Troubleshooting Oracle Fusion Middleware

This appendix provides information on how to troubleshoot problems that you might encounter when using Oracle Fusion Middleware. It contains the following sections:

- [Diagnosing Oracle Fusion Middleware Problems](#)
- [Common Problems and Solutions](#)
- [Troubleshooting SSL](#)
- [Need More Help?](#)

H.1 Diagnosing Oracle Fusion Middleware Problems

Oracle Fusion Middleware components generate log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, access information on HTTP requests, and additional information. The log files can be used to identify and diagnose problems. See [Chapter 11, "Managing Log Files and Diagnostic Data"](#) for more information about using and reading log files.

Oracle Fusion Middleware includes a Diagnostic Framework which aids in detecting, diagnosing, and resolving problems. The problems that are targeted in particular are critical errors such as those caused by code bugs, metadata corruption, and customer data corruption, deadlocked threads, and inconsistent state.

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error (such as log files) are immediately captured and tagged with this number. The data is then stored in the Automatic Diagnostic Repository (ADR), where it can later be retrieved by incident number and analyzed. See [Chapter 12, "Diagnosing Problems"](#) for more information about the Diagnostic Framework.

H.2 Common Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

- [Running out of Data Source Connections](#)
- [Using a Different Version of Spring](#)
- [ClassNotFoundException Errors When Starting Managed Servers](#)

H.2.1 Running out of Data Source Connections

If the database performance has slowed or you receive the following message in the Oracle WebLogic Server log files, you may have leaks in the data source connections:

```
No#resources#currently#available#in#pool#datasource name
#
Any#product#functionality#that#depend#on#the#datasource#will#not#function#as#it#
can't#connect#database#to#get#required#data.
```

If you receive this message, monitor the connection usage from the Administration Console data source monitoring page:

1. From Domain Structure, expand **Services**, then **Data Sources**.
2. Click the data source that you want to monitor.
3. Select the Monitoring tab, then the Statistics tab.
4. If the table does not display **Active Connection Current Count**, click **Customize this table**.
5. In Column Display, select **Active Connection Current Count** and move it from the Available to the Chosen box. Click **Apply**.
6. In the table, note the number in the **Active Connection Current Count** column.

If the active current count for a data source keeps increasing and does not go down, this data source is leaking connections. Contact Oracle Support.

H.2.2 Using a Different Version of Spring

When you configure a Managed Server with JRF, Spring 2.0.6 is installed and is placed in the Oracle WebLogic Server system classpath. If a custom application running in a JRF environment requires a different version of Spring, you must use the Filtering ClassLoader mechanism to specify the version of Spring.

Oracle WebLogic Server provides the FilteringClassLoader mechanism so that you can configure deployment descriptors to explicitly specify that certain packages should always be loaded from the application, rather than being loaded by the system classloader. This allows you to use alternate versions of applications such as Spring or Ant.

For more information about using the FilteringClassLoader mechanism, see "Using a Filtering ClassLoader" in the *Developing Applications for Oracle WebLogic Server*.

H.2.3 ClassNotFoundException Errors When Starting Managed Servers

If a Managed Server is started by Node Manager (as is the case when the servers are started by the Oracle WebLogic Server Administration Console or Fusion Middleware Control), you may receive a ClassNotFoundException error if Node Manager has not been configured to use the start scripts when starting Managed Servers. See [Section 2.8.1](#) for information about resolving this problem.

H.3 Troubleshooting SSL

This section describes common problems and solutions when working with SSL configuration. It contains the following topics:

- [Components May Enable All Supported Ciphers](#)
- [SSL Certificate Chain Required on Certain Browsers](#)

H.3.1 Components May Enable All Supported Ciphers

You should be aware that when no cipher is explicitly configured, some 12c (12.1.2) components enable all supported SSL ciphers including DH_Anon (Diffie-Hellman Anonymous) ciphers.

At this time, Oracle HTTP Server is the only component known to set ciphers like this. Configure the components with the desired cipher(s) if DH_Anon is not wanted.

H.3.2 SSL Certificate Chain Required on Certain Browsers

When you configure SSL for Oracle HTTP Server, you may need to import the entire certificate chain (root CA, Intermediate CAs and so on).

Certain browsers, for example Internet Explorer, require that the entire certificate chain be imported to the browsers for SSL handshake to work. If your certificate was issued by an intermediate CA, you will need to ensure that the complete chain of certificates is available on the browser or the handshake will fail. If an intermediate certificate in the chain expires, it must be renewed along with all the certificates (such as OHS server) in the chain.

H.4 Need More Help?

You can find more solutions on My Oracle Support, <http://support.oracle.com>. If you do not find a solution for your problem, log a service request.

You can also use the Remote Diagnostic Agent, as described in [Section H.4.1](#).

See Also: *Oracle Fusion Middleware Release Notes*, available on the Oracle Technology Network:

<http://docs.oracle.com/index.html#middleware>

H.4.1 Using Remote Diagnostic Agent

Remote Diagnostic Agent (RDA) is a command-line diagnostic tool that provides a comprehensive picture of your environment. Additionally, RDA can provide recommendations on various topics, for example configuration and security. This aids you and Oracle Support in resolving issues.

RDA is designed to be as unobtrusive as possible; it does not modify systems in any way. A security filter is provided if required.

For more information about RDA, see the readme file, which is located at:

```
(UNIX) #ORACLE_HOME/oracle_common/rda/README_Unix.txt  
(Windows) #ORACLE_HOME\oracle_common\rda\README_Windows.txt
```

