Oracle® Fusion Middleware

Administrator's Guide

11*g* Release 1 (11.1.1)

E10105-02

October 2009

Oracle Fusion Middleware Administrator's Guide, 11*g* Release 1 (11.1.1)

E10105-02

Primary Author: Helen Grembowicz

Contributing Author: Vinaye Misra

Contributors: Mike Blevins, Greg Cook, Shalendra Goel, Harry Hsu, Pavana Jain, Gopal Kirsur, Dan MacKinnon, Manoj Nayak, Mark Nelson, Sandeep Singh, Sunita Sharma

Contents

4 Starting and Stopping Oracle Fusion Middleware

5 Managing Ports

Part III Secure Sockets Layer

6 SSL Configuration in Oracle Fusion Middleware

7 Managing Keystores, Wallets, and Certificates

Part IV Deploying Applications

8 Understanding the Deployment Process

9 Deploying Applications

Part V Monitoring Oracle Fusion Middleware

10 Monitoring Oracle Fusion Middleware

11 Managing Log Files and Diagnostic Data

12 Diagnosing Problems

Part VI Advanced Administration

13 Managing the Metadata Repository

14 Changing Network Configurations

Part VII Advanced Administration: Backup and Recovery

15 Introducing Backup and Recovery

16 Backing Up Your Environment

17 Recovering Your Environment

Part VIII Advanced Administration: Expanding Your Environment

18 Scaling Your Environment

19 Cloning Oracle Fusion Middleware

20 Moving from a Test to a Production Environment

Part IX Appendixes

A Oracle Fusion Middleware Command-Line Tools

B URLs for Components

C Port Numbers

D Metadata Repository Schemas

E Using Oracle Fusion Middleware Accessibility Options

F Examples of Administrative Changes

G Viewing Release Numbers and Applying Patches

H Oracle Wallet Manager and orapki

I Troubleshooting Oracle Fusion Middleware

Index

List of Figures

List of Tables

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, and how to back up and recover Oracle Fusion Middleware.

Audience

This guide is intended for administrators of Oracle Fusion Middleware.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware 11*g* Release 1 (11.1.1) documentation set:

- *Oracle Fusion Middleware 2 Day Administration Guide*
- *Oracle Fusion Middleware Concepts*
- *Oracle Fusion Middleware Security Guide*
- *Oracle Fusion Middleware High Availability Guide*
- *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*
- *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*
- *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache*
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*

Conventions

The following text conventions are used in this document:

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

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.

New Features for Oracle Fusion Middleware 11*g* Release 1 (11.1.1.2)

Oracle Fusion Middleware 11*g* Release 1 (11.1.1.2) includes the following new features:

- The Diagnostic Framework, which aids in detecting, diagnosing, and resolving problems. For more information, see Chapter 12.
- An Oracle Common home, which contains the binary and library files required for the Oracle Enterprise Manager Fusion Middleware Control and Java Required Files (JRF). For more information, see Section 2.6.
- Changes to cloning. For more information, see Chapter 19.

New Features for Oracle Fusion Middleware 11*g* Release 1 (11.1.1)

Oracle Fusion Middleware 11*g* Release 1 (11.1.1) includes many new and changed features, including the following:

- The inclusion of Oracle WebLogic Server in Oracle Fusion Middleware, replacing Oracle Containers for Java EE. Oracle WebLogic Server is an enterprise-ready Java application server that supports the deployment of mission-critical applications in a robust, secure, highly available, and scalable environment. For an overview of Oracle WebLogic Server and the Oracle Fusion Middleware environment, see Section 2.1.
- New commands for many functions. Many components and services now use Oracle WebLogic Server Scripting Tool (WLST) commands. For example, commands to configure log files are WLST commands. See Section 3.5.1 for general information about invoking WLST.
- The Oracle Metadata Service (MDS) Repository, a particular type of repository that 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, such as Oracle B2B. For information about the MDS Repository, see Section 13.3.
- Wallet and Keystore Management: 11*g* Release 1 (11.1.1) provides new features for managing Oracle wallets and JKS keystores:
  - Password-protected wallets

When creating a wallet in prior releases, the administrator was always required to create a password-protected wallet. Once this wallet was created, the administrator could optionally create an auto-login wallet. Components needed the auto-login wallet at run time. Without an auto-login wallet, the password had to be specified in the component configuration file in an encrypted or obfuscated format.

In 11*g* Release 1 (11.1.1), this behavior has changed. Every time you create a password-protected wallet, an auto-login wallet is automatically created as well. This enables management tasks to be performed on the password-protected wallet, while components can use the auto-login wallet at run time. This eliminates the need to store passwords in configuration files.

To take advantage of this feature when creating a wallet with Fusion Middleware Control, you must uncheck the auto-login checkbox and enter the wallet password. Remember that this creates both the password-protected and auto-login wallets.

A new type of wallet has also been introduced, which is a standalone auto-login wallet. This wallet can be used for both management and run time without requiring a password. To create this wallet when creating a wallet with Fusion Middleware Control, check the auto-login check box. You do *not* need to provide a password for this type of wallet.

Note: The standalone auto-login wallet is the default choice for wallet creation.

– Wallet and Keystore Management Tools

In prior releases, Oracle Wallet Manager was the graphical interface tool and `orapki` the command-line tool to manage Oracle wallets.

In 11g Release 1 (11.1.1), you can use the Web-based interface Fusion Middleware Control or the WLST command-line tool to manage both Oracle wallets and JKS keystore files. One advantage of these new tools is that they allow you to manage keystores centrally across instances, because they work in the context of a management server.

While Oracle Wallet Manager is still available, its usage should be limited to PKCS#11 wallets (that is, Hardware Security Module integration).

You can still use `orapki` to manage both Oracle wallets and JKS keystores, but only local changes (on a per-instance basis) are possible. `orapki` allows you to manage PKCS#11 wallets and CRLs.

The following table shows the different tools and their capabilities:

Tool	Oracle Wallet	Java Keystore (JKS)	Local Updates	Distributed Updates	PKCS11	CRL	Graphical UI	Command Line
orapki (10*g*, 11*g*)	x	x	x		x	x		x
Oracle Wallet Manager (not in 11*g*)	x		x		x		x	
Fusion Middleware Control (new in 11*g*)	x	x		x			x	
WLST (new in 11*g*)	x	x		x				x

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 chapters:

- Chapter 1, "Introduction to Oracle Fusion Middleware"
- Chapter 2, "Understanding Oracle Fusion Middleware Concepts"

1

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 topics:

- 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, business intelligence, and collaboration. Oracle Fusion Middleware offers complete support for development, deployment, and management.

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. Oracle WebLogic Server is an ideal foundation for building applications based on service-oriented architecture (SOA).

  See Also: *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*

- Oracle SOA Suite, a complete set of service infrastructure components for designing, deploying, and managing composite applications. Oracle SOA Suite enables services to be created, managed, and orchestrated into composite applications and business processes. Composites enable you to easily assemble multiple technology components into one SOA composite application.

  See Also: *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite*

- Oracle WebCenter, an integrated set of components with which you can create social applications, enterprise portals, collaborative communities, and composite applications, built on a standards-based, service-oriented architecture. Oracle WebCenter combines dynamic user interface technologies with which to develop rich internet applications, the flexibility and power of an integrated, multichannel portal framework, and a set of horizontal Enterprise 2.0 capabilities delivered as services that provide content, collaboration, presence, and social networking capabilities. Based on these components, Oracle WebCenter also provides an out-of-the-box, enterprise-ready customizable application, WebCenter Spaces, with a configurable work environment that enables individuals and groups to work and collaborate more effectively.

> **See Also:** *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*

- 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:** *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server*

- Oracle Web Cache, a content-aware server accelerator, or reverse proxy, that improves the performance, scalability, and availability of Web sites that run on Oracle Fusion Middleware.

  > **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache*

- Oracle Identity Management, which provides a shared infrastructure for all Oracle applications. It also provides services and interfaces that facilitate third-party enterprise application development. These interfaces are useful for application developers who need to incorporate identity management into their applications.

  > **See Also:** *Oracle Fusion Middleware Integration Guide for Oracle Identity Management*

- Oracle Internet Directory, a general-purpose directory service that enables fast retrieval and centralized management of information about dispersed users and network resources. It combines Lightweight Directory Access Protocol (LDAP) Version 3 with the high performance, scalability, robustness, and availability of Oracle Database.

  > **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*

- Oracle Virtual Directory, an LDAP version 3 enabled service that provides virtualized abstraction of one or more enterprise data sources into a single directory view. Oracle Virtual Directory provides the ability to integrate LDAP-aware applications into diverse directory environments while minimizing or eliminating the need to change either the infrastructure or the applications. It supports a diverse set of clients, such as Web applications and portals, and it can connect to directories, databases, and Web Services.

  > **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*

- Oracle Identity Federation, a self-contained federation solution that provides the infrastructure that enables identities and their relevant entitlements to be propagated across security domains—this applies to domains existing within an organization as well as between organizations.

  See Also: *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation*

- 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: *Oracle Fusion Middleware Security and Administrator's Guide for 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: *Oracle Fusion Middleware Security Guide*

- Oracle Portal, a Web-based tool for building and deploying e-business portals. It provides a secure, manageable environment for accessing and interacting with enterprise software services and information resources. A portal page makes data from multiple sources accessible from a single location.

  See Also: *Oracle Fusion Middleware Administrator's Guide for Oracle Portal*

- Oracle Business Intelligence, a complete, integrated solution that addresses business intelligence requirements. Oracle Business Intelligence includes Oracle Business Intelligence Reporting and Publishing, Oracle Business Intelligence Discoverer, and Oracle Business Intelligence Publisher.

  See Also: *Oracle Fusion Middleware Configuration Guide for Oracle Business Intelligence Discoverer*

2 Understanding Oracle Fusion Middleware Concepts

This chapter provides information about Oracle Fusion Middleware concepts that are related to administering Oracle Fusion Middleware.

2.1 Understanding Key Oracle Fusion Middleware Concepts

Oracle Fusion Middleware provides two types of components:

- A **Java component,** which is an Oracle Fusion Middleware component that is deployed as one or more Java EE applications and a set of resources. Java components are deployed to an Oracle WebLogic Server domain as part of a domain template. Examples of Java components are the Oracle SOA Suite and Oracle WebCenter components.
- A **system component**, which is a manageable process that is not deployed as a Java application. Instead, a system component is managed by the Oracle Process Manager and Notification (OPMN). The system components are:
  - Oracle HTTP Server
  - Oracle Web Cache
  - Oracle Internet Directory
  - Oracle Virtual Directory
  - Oracle Forms Services
  - Oracle Reports
  - Oracle Business Intelligence Discoverer
  - Oracle Business Intelligence

A Java component and a system component are peers.

After you install and configure Oracle Fusion Middleware, your Oracle Fusion Middleware environment contains the following:

- An Oracle WebLogic Server domain, which contains one Administration Server and one or more Managed Servers. The Administration Server contains the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control. The Managed Servers contain components, such as Oracle WebCenter and Oracle SOA Suite.

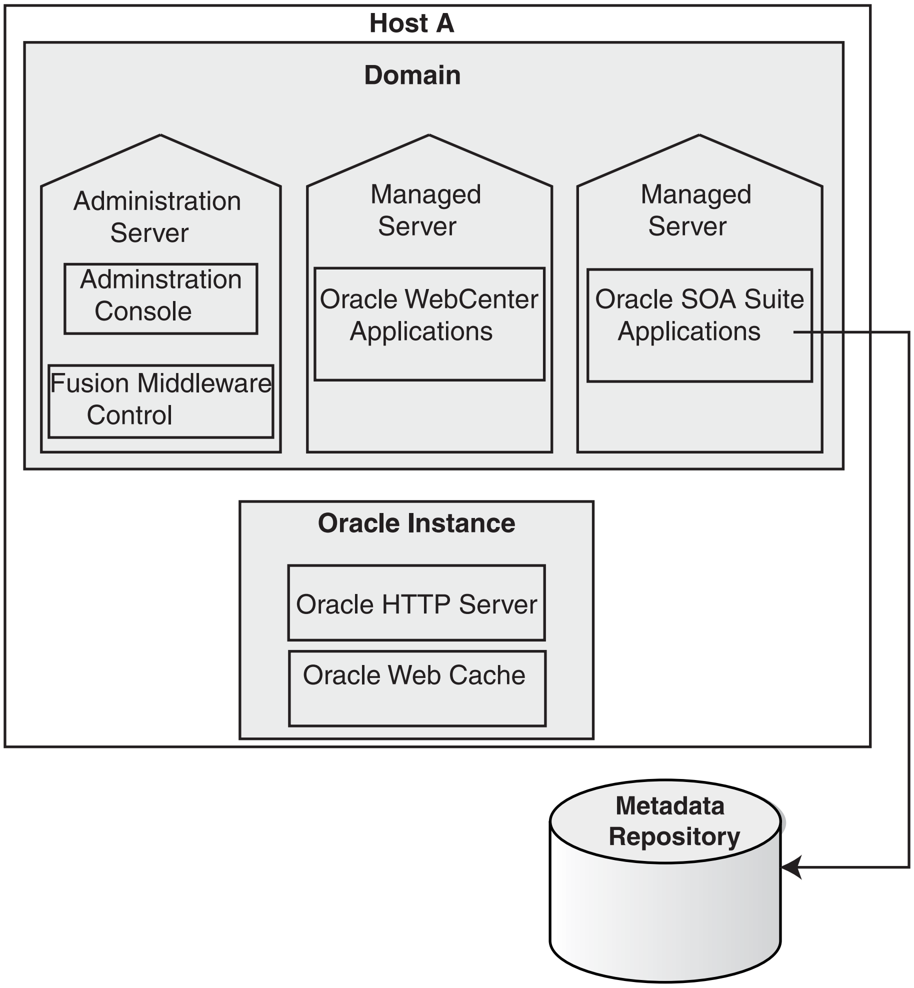
  See Section 2.2 for information about Oracle WebLogic Server domains.

- If your environment includes system components, one or more Oracle instances. See Section 2.3 for information about Oracle instances.

- A database that is used as a metadata repository, if the components you installed require one. For example, Oracle SOA Suite requires a metadata repository.

Figure 2–1 shows an Oracle Fusion Middleware environment with an Oracle WebLogic Server domain that contains an Administration Server, two Managed Servers, and an Oracle instance. The environment also includes a metadata repository.

Figure 2–1 Oracle Fusion Middleware Environment



Your environment also includes a Middleware home, which consists of the Oracle WebLogic Server home, and, optionally, an Oracle Common home and one or more Oracle homes. See Section 2.4.

2.2 What Is an Oracle WebLogic Server Domain?

An Oracle WebLogic Server administration **domain** is a logically related group of Java components. A domain includes a special WebLogic Server instance called the **Administration Server,** which is the central point from which you configure and manage all resources in the domain. Usually, you configure a domain to include additional WebLogic Server instances called **Managed Servers.** You deploy Java components, such as Web applications, EJBs, and Web services, and other resources to the Managed Servers and use the Administration Server for configuration and management purposes only.

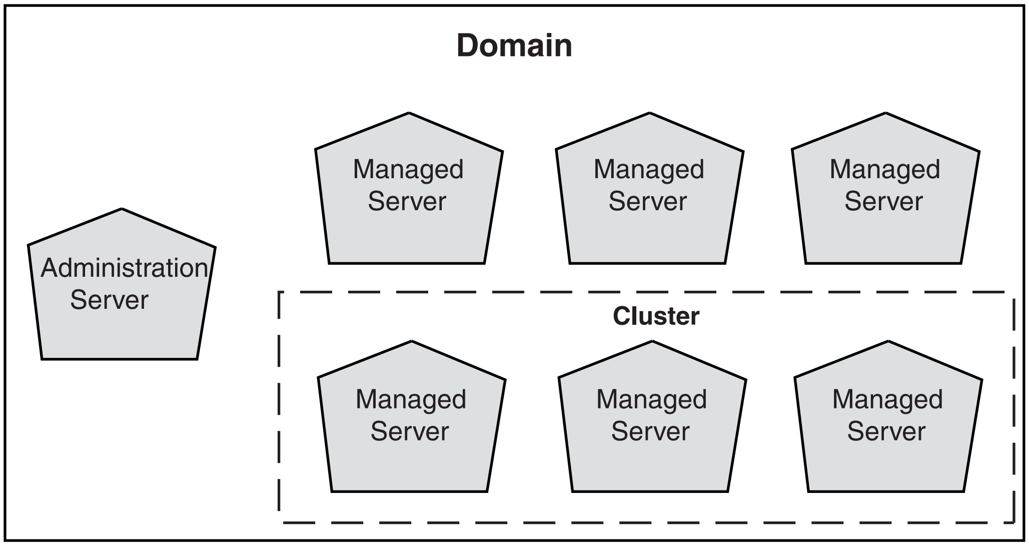
Managed Servers in a domain can be grouped together into a cluster.

The directory structure of a domain is separate from the directory structure of the WebLogic Server home. It can reside anywhere; it need not be within the Middleware home directory.

A domain is a peer of an Oracle instance. Both contain specific configurations outside of their Oracle homes.

Figure 2–2 shows a domain with an Administration Server, three standalone Managed Servers, and three Managed Servers in a cluster.

Figure 2–2 Oracle WebLogic Server Domain



See Also: *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server* for more information about domain configuration

The following topics describe entities in the domain:

- What Is the Administration Server?
- Understanding Managed Servers and Managed Server Clusters
- What Is Node Manager?

2.2.1 What Is the Administration Server?

The **Administration Server** operates as the central control entity for the configuration of the entire domain. It maintains the domain's configuration documents and distributes changes in the configuration documents to Managed Servers. The Administration Server serves as a central location from which to monitor all resources in a domain.

Each domain must have one server instance that acts as the Administration Server.

To interact with the Administration Server, you can use the Oracle WebLogic Server Administration Console, Oracle WebLogic Scripting Tool (WLST), or create your own JMX client. In addition, you can use Oracle Enterprise Manager Fusion Middleware Control for some tasks.

Oracle WebLogic Server Administration Console and Fusion Middleware Control run in the Administration Server. Oracle WebLogic Server Administration Console is the Web-based administration console used to manage the resources in an Oracle WebLogic Server domain, including the Administration Server and Managed Servers. Fusion Middleware Control is a Web-based administration console used to manage Oracle Fusion Middleware, including components such as Oracle HTTP Server, Oracle SOA Suite, Oracle WebCenter, Oracle Portal, and Oracle Identity Management.

See Also:

- Section 3.3 for more information about Fusion Middleware Control
- Section 3.4 of this book, as well as the *Oracle Fusion Middleware Introduction to Oracle WebLogic Server* and the Oracle WebLogic Server Administration Console Online help, for more information about Oracle WebLogic Server Administration Console

2.2.2 Understanding Managed Servers and Managed Server Clusters

Managed Servers host business applications, application components, Web services, and their associated resources. To optimize performance, Managed Servers maintain a read-only copy of the domain's configuration document. When a Managed Server starts, it connects to the domain's Administration Server to synchronize its configuration document with the document that the Administration Server maintains.

When you create a domain, you create it using a particular domain template. That template supports a particular component or group of components, such as the Oracle SOA Suite. The Managed Servers in the domain are created specifically to host those particular Oracle Fusion Middleware components.

Oracle Fusion Middleware Java components (such as Oracle SOA Suite, Oracle WebCenter, and some Identity Management components), as well as customer-developed applications, are deployed to Managed Servers in the domain.

If you want to add other components, such as Oracle WebCenter, to a domain that was created using a template that supports another component, you can extend the domain by creating additional Managed Servers in the domain, using a domain template for the component that you want to add. See Section 18.2 for more information.

For production environments that require increased application performance, throughput, or high availability, you can configure two or more Managed Servers to operate as a cluster. A **cluster** is a collection of multiple Oracle WebLogic Server server instances running simultaneously and working together to provide increased scalability and reliability. In a cluster, most resources and services are deployed identically to each Managed Server (as opposed to a single Managed Server), enabling failover and load balancing. A single domain can contain multiple Oracle WebLogic Server clusters, as well as multiple Managed Servers that are not configured as clusters. The key difference between clustered and nonclustered Managed Servers is support for failover and load balancing. These features are available only in a cluster of Managed Servers.

> **See Also:** "Understanding WebLogic Server Clustering" in *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*

2.2.3 What Is Node Manager?

Node Manager is a Java utility that runs as a separate process from Oracle WebLogic Server and allows you to perform common operations for a Managed Server, regardless of its location with respect to its Administration Server. While use of Node Manager is optional, it provides valuable benefits if your Oracle WebLogic Server environment hosts applications with high-availability requirements.

If you run Node Manager on a computer that hosts Managed Servers, you can start and stop the Managed Servers remotely using the Administration Console or the command line. Node Manager can also automatically restart a Managed Server after an unexpected failure.

> **See Also:** *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server*

2.3 What Is an Oracle Instance?

An **Oracle instance** contains one or more system components, such as Oracle Web Cache, Oracle HTTP Server, or Oracle Internet Directory. The system components in an Oracle instance must reside on the same computer. An Oracle instance directory contains updatable files, such as configuration files, log files, and temporary files.

An Oracle instance is a peer of an Oracle WebLogic Server domain. Both contain specific configurations outside of their Oracle homes.

The directory structure of an Oracle instance is separate from the directory structure of the Oracle home. It can reside anywhere; it need not be within the Middleware home directory.

2.4 What Is a Middleware Home?

A **Middleware home** is a container for the Oracle WebLogic Server home, and, optionally, one Oracle Common home and one or more Oracle homes.

A Middleware home can reside on a local file system or on a remote shared disk that is accessible through NFS.

See Section 2.5 for information about Oracle WebLogic Server homes. See Section 2.6 for information about Oracle homes.

2.5 What Is a WebLogic Server Home?

A WebLogic Server home contains installed files necessary to host a WebLogic Server. The WebLogic Server home directory is a peer of Oracle home directories and resides within the directory structure of the Middleware home.

2.6 What Is an Oracle Home and the Oracle Common Home?

An **Oracle home** contains installed files necessary to host a specific component or software suite. For example, the SOA Oracle home contains a directory that contains binary and library files for Oracle SOA Suite.

An Oracle home resides within the directory structure of the Middleware home. Each Oracle home can be associated with multiple Oracle instances or Oracle WebLogic Server domains. There can be multiple Oracle homes within each Middleware home.

The **Oracle Common home** contains the binary and library files required for the Oracle Enterprise Manager Fusion Middleware Control and Java Required Files (JRF). There can be only one Oracle Common home within each Middleware home.

Part II

Basic Administration

This part describes basic administration tasks.

Part II contains the following chapters:

- Chapter 3, "Getting Started Managing Oracle Fusion Middleware"
- Chapter 4, "Starting and Stopping Oracle Fusion Middleware"
- Chapter 5, "Managing Ports"

3 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 topics:

- Setting Up Environment Variables
- 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 Command-Line Tools
- Getting Started Using the Fusion Middleware Control MBean Browsers
- Managing Components
- Changing the Administrative User Password
- Basic Tasks for Configuring and Managing Oracle Fusion Middleware

3.1 Setting Up Environment Variables

When you installed Oracle Fusion Middleware, you were logged in to your operating system as a particular user. You should always log in as this user to manage your installation because this user has permission to view and modify the files in your installation's Oracle home.

To use Oracle Fusion Middleware, you must set environment variables as shown in the following tables:

- Table 3–1, " Environment Variables for Linux and UNIX"
- Table 3–2, " Environment Variables for Windows"

Table 3–1 Environment Variables for Linux and UNIX

Environment Variable	Value
DISPLAY	`hostname:display_number.screen_number` Beginning with Oracle Application Server 10*g*, very few tools, such as `oidadmin`, require the DISPLAY variable.

Table 3–1 (Cont.) Environment Variables for Linux and UNIX

Environment Variable	Value

LD_LIBRARY_PATH	On Solaris, ensure that the value contains the following directory:
	`$ORACLE_HOME/lib32`
	On Linux and HP-UX, ensure that the value contains the following directory:
	`$ORACLE_HOME/lib`
	On IBM AIX, ensure that this environment variable is not set.

(IBM AIX only)
LIBPATH
	If the calling application is a 32-bit application, ensure that the value contains the following directory:
	`$ORACLE_HOME/lib32`
	If the calling application is a 64-bit application, ensure that the value contains the following directory:
	`$ORACLE_HOME/lib`

(Solaris only) LD_LIBRARY_PATH_64	Ensure that the value contains the following directory: `$ORACLE_HOME/lib`
(HP-UX only) SHLIB_PATH	Ensure that the value contains the following directory: `$ORACLE_HOME/lib32`
MW_HOME	Set the value to the full path of the installation's Middleware home. Do not use a trailing slash in the definition. The following example shows the full path: `/scratch/Oracle/Middleware`
ORACLE_HOME	Setting this is useful if you have only one Oracle home in your environment or if you will be working with just that one Oracle home. Set to the full path of the Oracle home. Do not use a trailing slash in the definition. The following example shows the full path: `/scratch/Oracle/Middleware/ORACLE_HOME_SOA1`
ORACLE_INSTANCE	Optional. Setting this is useful if you have only one Oracle instance in your environment or if you will be working with just that one instance. Set to the full path of an Oracle instance. Do not use a trailing slash in the definition. The following example shows the full path of a Web Tier installation: `scratch/Oracle/Middleware/WebTier/instances/instance1`
PATH	Ensure that the value contains the following directory, which contains basic commands used by all installations: `$ORACLE_COMMON_HOME/bin` When you start to work with specific components, you may want to add additional directories to your path, as recommended by the component documentation.
JAVA_HOME	Ensure that the value contains the following directory: *MW_HOME*/jdk*n*
CLASSPATH	Ensure that the value contains the following directories: $ORACLE_HOME/lib:*MW_HOME*/jdk*n*/lib

Table 3–2 shows the environment variables for Windows.

Table 3–2 Environment Variables for Windows

Environment Variable	Value
MW_HOME	Set the value to the full path of the installation's Middleware home. Do not use a trailing slash in the definition. The following example shows the full path: `C:\oracle\Middleware`
ORACLE_HOME	Setting this is useful if you have only one Oracle home in your environment or if you will be working with just that one Oracle home. Set the value to the full path of the Oracle home. Do not use a trailing backslash in the definition. The following example shows the full path: `C:\oracle\Middleware\ORACLE_COMMON_HOME` The value is automatically set by Oracle Universal Installer.
ORACLE_INSTANCE	Optional. Setting this is useful if you have only one Oracle instance in your environment or if you will be working with just that one instance. Set the value to the full path of an Oracle instance. Do not use a trailing backslash in the definition. The following example shows the full path of a Web Tier installation: `C:\oracle\Middleware\WebTier\instances\instance1`
PATH	Ensure that the value contains the following directory, which contains basic commands used by all installations: *ORACLE_COMMON_HOME*`\bin`
JAVA_HOME	Ensure that the value contains the following directory: *MW_HOME*`\jdk`*n*
CLASSPATH	Ensure that the value contains the following directories: *ORACLE_HOME*`\lib:`*MW_HOME*`\jdk`*n*`\lib`
TEMP	Set the value to your temp directory, for example, C:\temp.
TMP	Set the value to your temp directory, for example, C:\temp.

Best Practices for Multiple Installations on a UNIX Host

If you have multiple installations of Oracle Fusion Middleware on a UNIX host, it is very important to completely set your environment when managing a particular installation.

Some Oracle Fusion Middleware commands use the MW_HOME and ORACLE_HOME environment variables to determine which installation to operate on, and some use the directory location of the command. It is, therefore, not sufficient to simply reset your environment variables or change directories to a different Oracle home as you move between installations. You must fully change to the new installation as follows:

1. Log in as the user who installed Oracle Fusion Middleware.

   On UNIX hosts, you may also use the `su` command to switch to the user, but be sure to use the dash (`-`) option so that your environment is set the same as it would have been had you actually logged in as that user. For example:

   ```
   su - user
   ```

2. Set the correct environment variables for the installation, as described in Table 3–1.

3. Execute commands in the Middleware home and Oracle home of the correct installation.

Multiple Installations by the Same User If you installed multiple installations as the same user, ensure that you are in the correct Middleware home and Oracle home and have the correct environment variables set when working on a particular installation. You may want to set up some scripts to make it easy to change from one installation to another.

3.2 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 3.3.
- Oracle WebLogic Server Administration Console. See Section 3.4
- The Oracle Fusion Middleware command-line tools. See Section 3.5.
- The Fusion Middleware Control MBean Browser. See Section 3.6.

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 perform some tasks with either tool, but for other tasks, you can only use one of the tools. Table 3–3 lists some common tasks with the recommended tool.

Table 3–3 Comparing Fusion Middleware Control and WebLogic Server Administration Console

Task	Tool to Use
Manage Oracle WebLogic Server	**Use:**
Create additional Managed Servers	WebLogic Server Administration Console
Clone Managed Servers	WebLogic Server Administration Console
Cluster Managed Servers	WebLogic Server Administration Console
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 or use the LDAP server's tool when using another LDAP server
Manage Data Sources	**Use:**
Create data sources	WebLogic Server Administration Console
Create connection pools	WebLogic Server Administration Console
Manage JMS Resources	**Use:**
Create JMS queues	WebLogic Server Administration Console
Configure advanced queuing	WebLogic Server Administration Console
Manage SOA environment	**Use:**
Deploy SOA Composite applications	Fusion Middleware Control

Table 3–3 (Cont.) Comparing Fusion Middleware Control and WebLogic Server Administration Console

Task	Tool to Use
Monitor SOA Composite applications	Fusion Middleware Control
Modify Oracle BPEL Process Manager MBean properties	Fusion Middleware Control
Debug applications such as Oracle BPEL Process Manager applications	Fusion Middleware Control
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:**

View and manage log files	Fusion Middleware Control for most log files
	WebLogic Server Administration Console for the following logs:
	`DOMAIN_HOME/servers/server_name/logs/access.log`
	`DOMAIN_HOME/servers/server_name/data/ldap/log/EmbeddedLDAP.log`
	`DOMAIN_HOME/servers/server_name/data/ldap/log/EmbeddedLDAPAccess.log`

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
Manage Oracle Web Cache	Fusion Middleware Control
Start and stop components	Fusion Middleware Control
Start and stop applications	Fusion Middleware Control

3.3 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 a farm.

A **farm** is a collection of components managed by Fusion Middleware Control. It can contain Oracle WebLogic Server domains, one Administration Server, one or more

Managed Servers, clusters, and the Oracle Fusion Middleware components that are installed, configured, and running in the domain.

Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the farm, 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 Farm
- Viewing and Managing Components
- Viewing the Status of Applications

3.3.1 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 port number assigned during the installation. The following shows the format of the URL:

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

For some installation types, such as the 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, such as Oracle SOA Suite, the information is displayed on the Create Domain screen of the Configuration Wizard when the configuration completes.

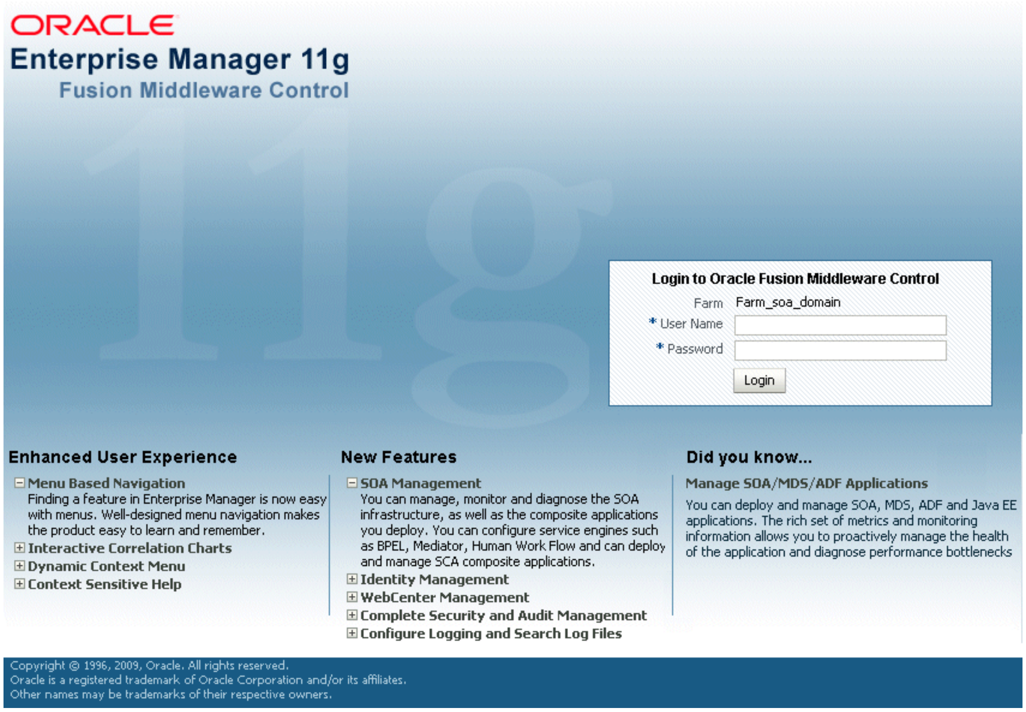
The port number is the number of the Administration Server. By default, the port number is 7001.

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:



ORACLE Enterprise Manager 11g
Fusion Middleware Control

Login to Oracle Fusion Middleware Control

Farm Farm_soa_domain

* User Name

* Password

Enhanced User Experience

⊟ Menu Based Navigation
Finding a feature in Enterprise Manager is now easy with menus. Well-designed menu navigation makes the product easy to learn and remember.

⊞ Interactive Correlation Charts

⊞ Dynamic Context Menu

⊞ Context Sensitive Help

New Features

⊟ SOA Management
You can manage, monitor and diagnose the SOA infrastructure, as well as the composite applications you deploy. You can configure service engines such as BPEL, Mediator, Human Work Flow and can deploy and manage SCA composite applications.

⊞ Identity Management

⊞ WebCenter Management

⊞ Complete Security and Audit Management

⊞ Configure Logging and Search Log Files

Did you know...

Manage SOA/MDS/ADF Applications

You can deploy and manage SOA, MDS, ADF and Java EE applications. The rich set of metrics and monitoring information allows you to proactively manage the health of the application and diagnose performance bottlenecks

Copyright © 1996, 2009, Oracle. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

2. Enter the 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.

3.3.2 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. In most cases, the Help window displays a help topic about the current page. Click **Contents** in the Help window to browse the list of help topics, or click **Search** to search for a particular word or phrase.

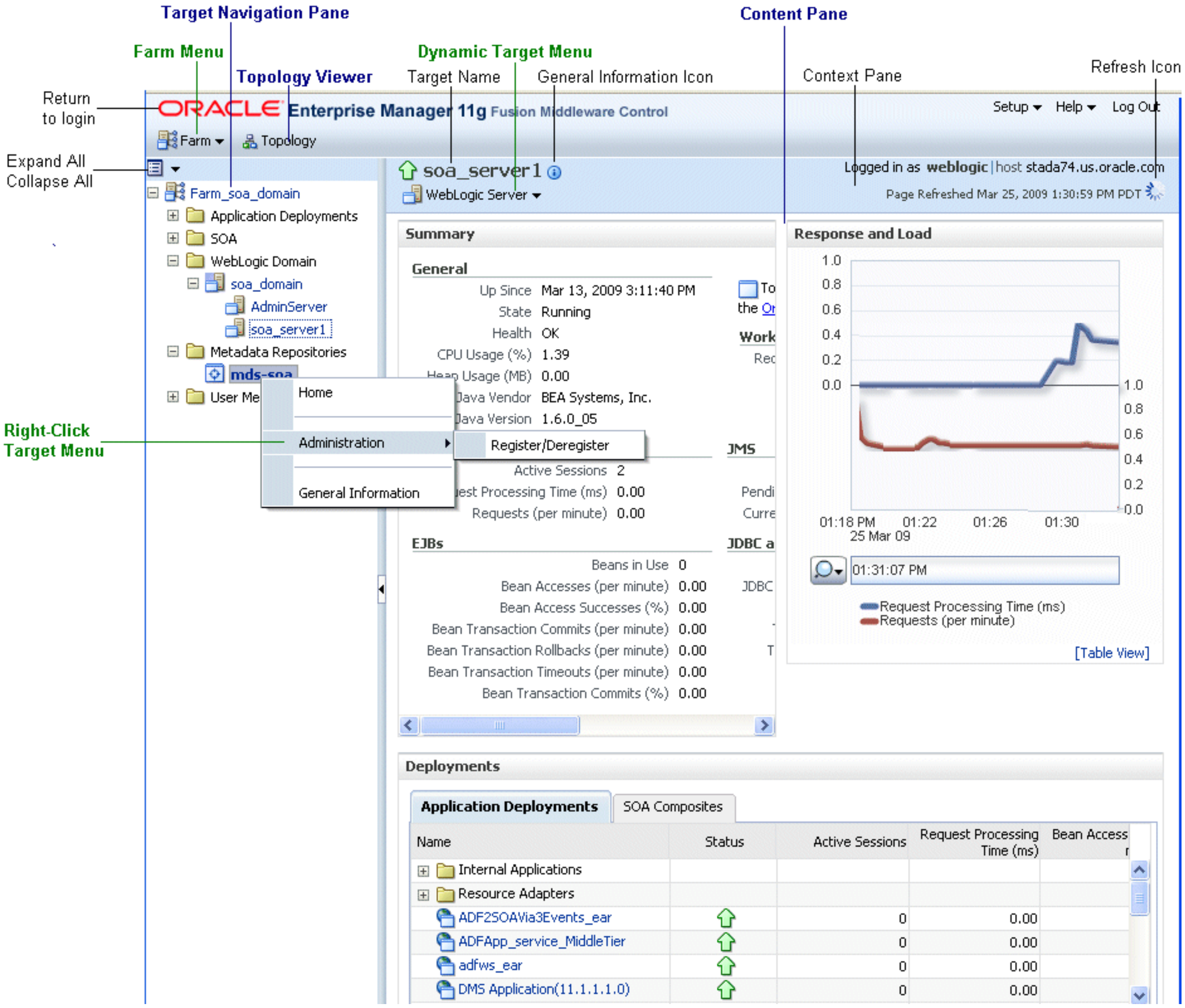
3.3.3 Navigating Within Fusion Middleware Control

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 farm 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.



General

Up Since	Mar 13, 2009 3:11:40 PM
State	Running
Health	OK
CPU Usage (%)	1.39
Heap Usage (MB)	0.00
Java Vendor	BEA Systems, Inc.
Java Version	1.6.0_05
Active Sessions	2
Request Processing Time (ms)	0.00
Requests (per minute)	0.00

EJBs

Beans in Use	0
Bean Accesses (per minute)	0.00
Bean Access Successes (%)	0.00
Bean Transaction Commits (per minute)	0.00
Bean Transaction Rollbacks (per minute)	0.00
Bean Transaction Timeouts (per minute)	0.00
Bean Transaction Commits (%)	0.00

Name	Status	Active Sessions	Request Processing Time (ms)	Bean Access
Internal Applications				
Resource Adapters				
ADF2SOAVia3Events_ear	⇧	0	0.00	
ADFApp_service_MiddleTier	⇧	0	0.00	
adfws_ear	⇧	0	0.00	
DMS Application(11.1.1.1.0)	⇧	0	0.00	

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

- **Target Navigation Pane** lists all of the targets in the farm 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.
- **Farm Menu** provides a list of operations that you can perform on the farm. The Farm menu is always available.
- **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.**

- **Topology Viewer** displays the topology of the farm.
- **Target Name** is the name of the currently selected target.
- **General 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.
- **Expand All/Collapse All** lets you expand or collapse the navigation tree.
- **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.)
- **Return to login** takes you to the login page when you click the Oracle Enterprise Manager logo.

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. For information about configuring single sign-on between Fusion Middleware Control and the WebLogic Server Administration Console, see "Configuring Single Sign-On for Administration Consoles" in the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management.*

Table 3–4 describes some common ways you can navigate within Fusion Middleware Control.

Table 3–4 Navigating Within Fusion Middleware Control

To:	Take This Action:
View all of the targets in the farm	Click the **Expand All** icon at the top of the **target navigation pane**.
Navigate to the farm	Select the farm from the **target navigation pane**. The farm's home page is displayed in the content pane.
Operate on the farm	Select the **Farm 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: *logfile_name*
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 General Information icon.
Return to the login page	Click the **Oracle Enterprise Manager logo** at the top left of the page.

Table 3–4 (Cont.) Navigating Within Fusion Middleware Control

To:	Take This Action:
View the topology	Click **Topology.**
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.

3.3.4 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.

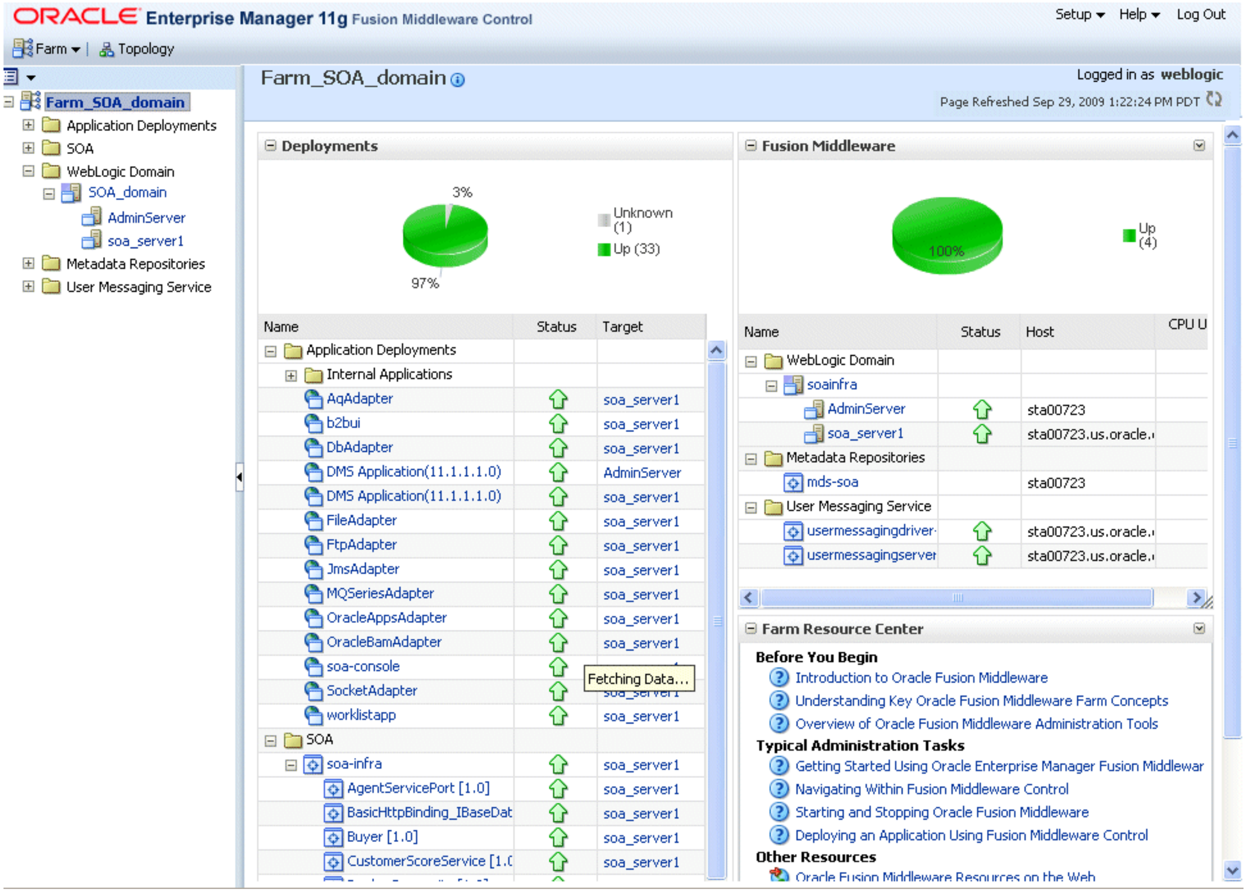
Each role defines the type of access a user has. For example, a user with the role Admin has full privileges. A user with the role Operator has privileges to perform essential day-to-day operations. A user with the role Monitor has privileges only to view the configuration.

> **See Also:** "Users, Groups, and Security Roles" in the *Oracle Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server*

3.3.5 Viewing and Managing the Farm

When you log in to Fusion Middleware Control, the first page you see is the Farm home page. You can also view this page at any time by selecting the farm in the target navigation pane.

The following figure shows the Farm home page:



Name	Status	Target
Application Deployments		
Internal Applications		
AqAdapter	⇧	soa_server1
b2bui	⇧	soa_server1
DbAdapter	⇧	soa_server1
DMS Application(11.1.1.1.0)	⇧	AdminServer
DMS Application(11.1.1.1.0)	⇧	soa_server1
FileAdapter	⇧	soa_server1
FtpAdapter	⇧	soa_server1
JmsAdapter	⇧	soa_server1
MQSeriesAdapter	⇧	soa_server1
OracleAppsAdapter	⇧	soa_server1
OracleBamAdapter	⇧	soa_server1
soa-console	⇧	
SocketAdapter	⇧	soa_server1
worklistapp	⇧	soa_server1
SOA		
soa-infra	⇧	soa_server1
AgentServicePort [1.0]	⇧	soa_server1
BasicHttpBinding_IBaseDat	⇧	soa_server1
Buyer [1.0]	⇧	soa_server1
CustomerScoreService [1.(	⇧	soa_server1

Name	Status	Host	CPU U
WebLogic Domain			
soainfra			
AdminServer	⇧	sta00723	
soa_server1	⇧	sta00723.us.oracle.(	
Metadata Repositories			
mds-soa		sta00723	
User Messaging Service			
usermessagingdriver-	⇧	sta00723.us.oracle.(	
usermessagingserver	⇧	sta00723.us.oracle.(	

Farm Resource Center

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

Typical Administration Tasks
- Getting Started Using Oracle Enterprise Manager Fusion Middlewar
- Navigating Within Fusion Middleware Control
- Starting and Stopping Oracle Fusion Middleware
- Deploying an Application Using Fusion Middleware Control

Other Resources
- Oracle Fusion Middleware Resources on the Web

The Farm menu is displayed at the top of the page. From the Farm menu, you can take the following actions:

- Create clusters.
- View log messages.
- Specify monitoring credentials

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

You can view the farm topology by selecting **Topology**. 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.

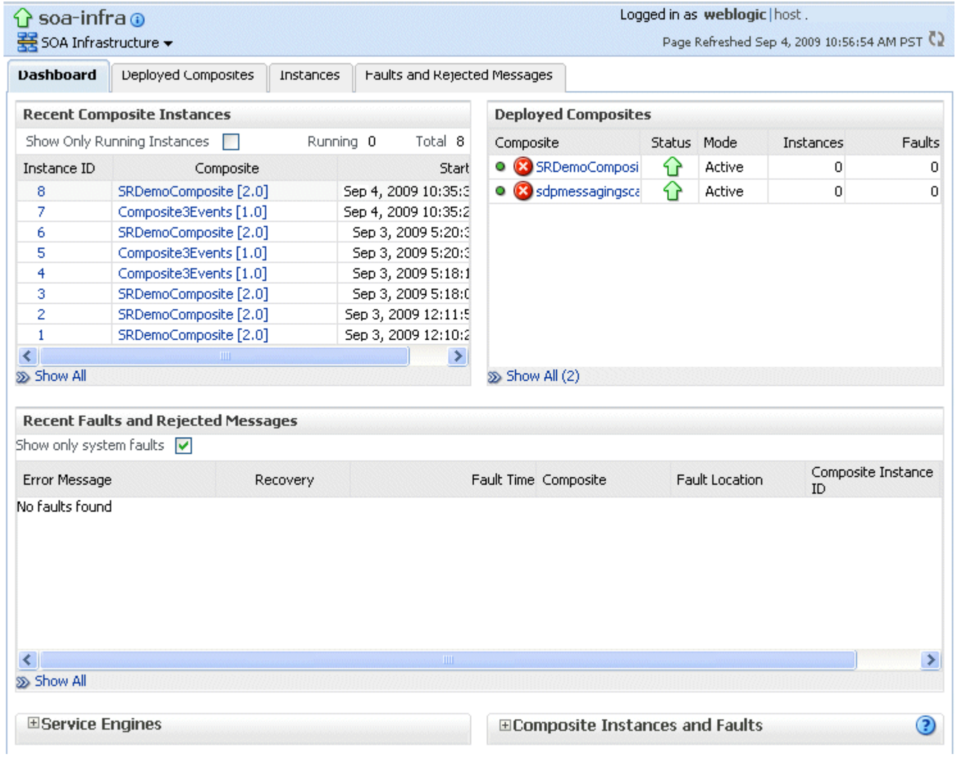
3.3.6 Viewing and Managing Components

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

For example, to view and manage Oracle SOA Suite, take the following steps:

1. In the target navigation pane, expand the farm, then **SOA.**
2. Select the SOA instance.

   The home page for the SOA instance is displayed, as shown in the following figure:



soa-infra

SOA Infrastructure ▾

Logged in as **weblogic** | host .

Page Refreshed Sep 4, 2009 10:56:54 AM PST

Recent Composite Instances

Show Only Running Instances □ Running 0 Total 8

Instance ID	Composite	Start
8	SRDemoComposite [2.0]	Sep 4, 2009 10:35:3
7	Composite3Events [1.0]	Sep 4, 2009 10:35:2
6	SRDemoComposite [2.0]	Sep 3, 2009 5:20:3
5	Composite3Events [1.0]	Sep 3, 2009 5:20:3
4	Composite3Events [1.0]	Sep 3, 2009 5:18:1
3	SRDemoComposite [2.0]	Sep 3, 2009 5:18:0
2	SRDemoComposite [2.0]	Sep 3, 2009 12:11:5
1	SRDemoComposite [2.0]	Sep 3, 2009 12:10:2

Show All

Deployed Composites

Composite	Status	Mode	Instances	Faults
● ☒ SRDemoComposi	⇧	Active	0	0
● ☒ sdpmessagingsca	⇧	Active	0	0

Show All (2)

Recent Faults and Rejected Messages

Show only system faults ☑

Error Message	Recovery	Fault Time	Composite	Fault Location	Composite Instance ID
No faults found					

Show All

⊞Service Engines

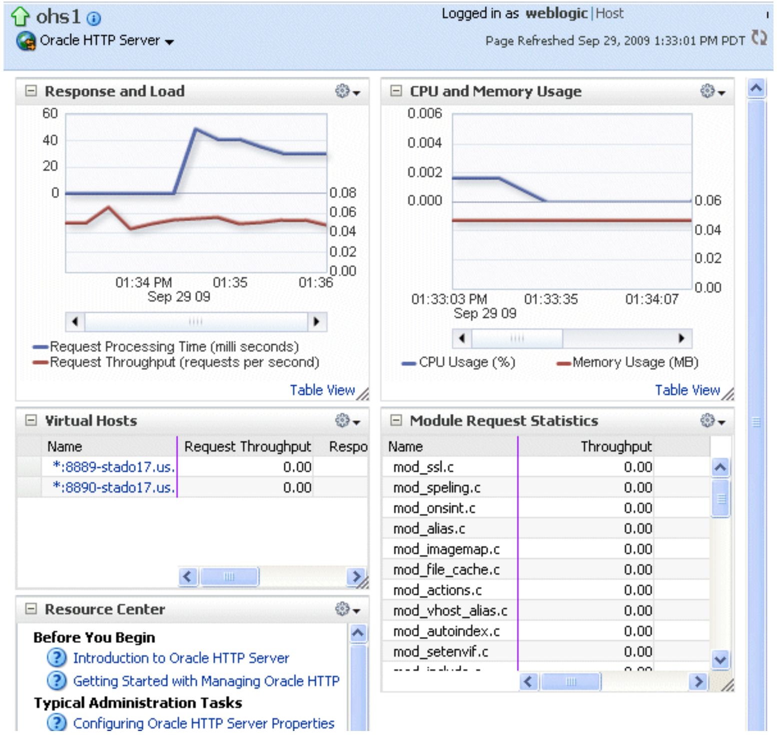
⊞Composite Instances and Faults

3. From the SOA Infrastructure menu, you can perform many administrative tasks, such as starting, stopping, and monitoring Oracle SOA Suite and deploying SOA composite applications.

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

1. In the target navigation pane, expand the farm, then **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:



Name	Request Throughput	Respo
*:8889-stado17.us.	0.00	
*:8890-stado17.us.	0.00	

Name	Throughput
mod_ssl.c	0.00
mod_speling.c	0.00
mod_onsint.c	0.00
mod_alias.c	0.00
mod_imagemap.c	0.00
mod_file_cache.c	0.00
mod_actions.c	0.00
mod_vhost_alias.c	0.00
mod_autoindex.c	0.00
mod_setenvif.c	0.00

Before You Begin

- Introduction to Oracle HTTP Server
- Getting Started with Managing Oracle HTTP

Typical Administration Tasks

- Configuring Oracle HTTP Server Properties

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

> **See Also:** Section 10.1.5 for more information about monitoring components

3.3.7 Viewing the Status of Applications

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

To view Java EE applications:

1. From the target navigation pane, expand the farm and then **Application Deployments.**
2. Select the application that you want to view.

   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.

To view SOA Composite Applications:

1. From the target navigation pane, expand the farm, then **SOA,** and then **soa-infra.**
2. Select the application that you want to view.

   The application's home page is displayed. It shows information about the application, such as the recent instances of the application, the faults and rejected messages and the policies.

3.4 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

3.4.1 Displaying the Oracle WebLogic Server Administration Console

To display the Administration Console:

1. Enter the following URL in a browser:

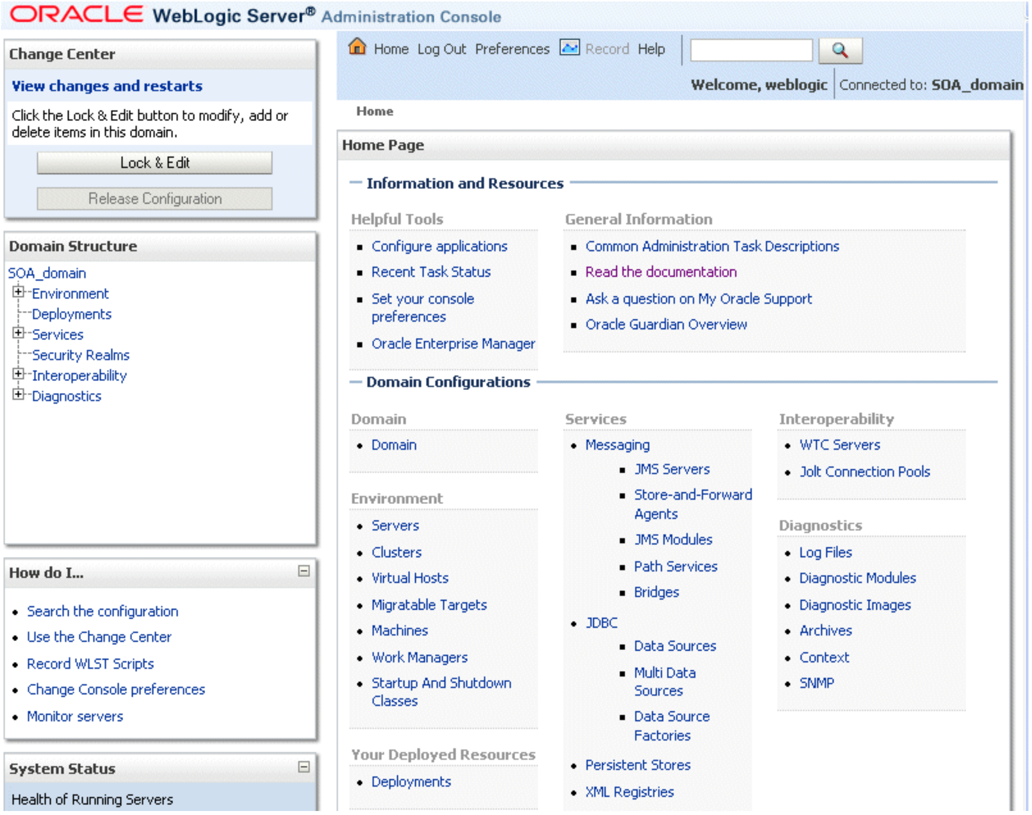
   ```
   http://hostname:port_number/console
   ```

   The port number is the 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:



ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic | Connected to: SOA_domain

Change Center

View changes and restarts

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

Lock & Edit

Release Configuration

Domain Structure

SOA_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...
- Search the configuration
- Use the Change Center
- Record WLST Scripts
- Change Console preferences
- Monitor servers

System Status

Health of Running Servers

Home

Home Page

Information and Resources

Helpful Tools
- Configure applications
- Recent Task Status
- Set your console preferences
- Oracle Enterprise Manager

General Information
- Common Administration Task Descriptions
- Read the documentation
- Ask a question on My Oracle Support
- Oracle Guardian Overview

Domain Configurations

Domain
- Domain

Environment
- Servers
- Clusters
- Virtual Hosts
- Migratable Targets
- Machines
- Work Managers
- Startup And Shutdown Classes

Your Deployed Resources
- Deployments

Services
- Messaging
  - JMS Servers
  - Store-and-Forward Agents
  - JMS Modules
  - Path Services
  - Bridges
- JDBC
  - Data Sources
  - Multi Data Sources
  - Data Source Factories
- Persistent Stores
- XML Registries

Interoperability
- WTC Servers
- Jolt Connection Pools

Diagnostics
- Log Files
- Diagnostic Modules
- Diagnostic Images
- Archives
- Context
- SNMP

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.

3.4.2 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 the pending changes.

3.5 Getting Started Using Command-Line Tools

Oracle Fusion Middleware provides the following primary command-line tools to manage most Oracle Fusion Middleware components:

- WebLogic Scripting Tool (WLST). See Section 3.5.1.

- Oracle Process Manager and Notification Server (OPMN). See Section 3.5.2.

3.5.1 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.

You can use WLST commands 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:

```
java weblogic.WLST
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 soa_server1, use the following command:

```
wls:/SOA_domain/serverConfig> state('soa_server1', 'Server')
Current state of 'soa_server1' : RUNNING
```

> **See Also:** *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

3.5.1.1 Using Custom WLST Commands

Many components, such as Oracle SOA Suite, Oracle Platform Security Services (OPSS), Oracle Fusion Middleware Audit Framework, and MDS, and services such as SSL and logging, supply custom WLST commands.

To use those custom commands, you must invoke the WLST script from the Oracle home in which the component has been installed. Do not use the WLST script in the WebLogic Server home.

For the following components and services, invoke WLST from the Oracle Common home:

- Oracle Application Development Framework
- Oracle Fusion Middleware Audit Framework
- Oracle Access Manager
- Oracle Platform Security Services
- Oracle Metadata Service
- Diagnostic Framework
- Dynamic Monitoring Service (DMS)
- Logging
- Secure Sockets Layer (SSL)
- Oracle Web Services
- Oracle Web Services Manager

The script is located at:

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

For other components, such as Oracle HTTP Server, Oracle SOA Suite, or Oracle WebCenter, invoke WLST from the Oracle home in which the component has been installed. The script is located at:

```
(UNIX) ORACLE_HOME_for_component/common/bin/wlst.sh
(Windows) ORACLE_HOME_for_component\common\bin\wlst.cmd
```

For example, to run the custom WLST commands for Oracle SOA Suite on a Linux system, use the following commands:

```
cd SOA_ORACLE_HOME/common/bin
./wlst.sh
```

3.5.1.2 Using WLST Commands for System Components

In addition to the commands provided by WLST for Oracle WebLogic Server, WLST provides a subset of commands to manage system components. These commands are:

- startproc(*componentName* [, *componentType*] [, *componentSet*): Starts the specified component
- stopproc(*componentName* [, *componentType*] [, *componentSet*): Stops the specified component
- status(*componentName* [, *componentType*] [, *componentSet*): Obtains the status of the specified component
- proclist(): Obtain the list of components

To use these custom commands, you must invoke the WLST script from the Oracle 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_for_component/common/bin/wlst.sh
(Windows) ORACLE_HOME_for_component\common\bin\wlst.cmd
```

3.5.2 Getting Started Using Oracle Process Manager and Notification Server

Oracle Process Manager and Notification Server (OPMN) manages and monitors the following Oracle Fusion Middleware components, referred to as system components:

- Oracle HTTP Server
- Oracle Web Cache
- Oracle Internet Directory
- Oracle Virtual Directory
- Oracle Forms Services
- Oracle Reports
- Oracle Business Intelligence Discoverer
- Oracle Business Intelligence

OPMN provides the `opmnctl` command. The executable file is located in the following directory, which you should add to your PATH environment variable:

```
(UNIX) ORACLE_HOME/opmn/bin
(Windows) ORACLE_HOME\opmn\bin
```

To view the status of all system components in an Oracle instance, use the following command:

```
opmnctl status
Processes in Instance: webtier_inst
---------------------------------+--------------------+---------+---------
ias-component                    | process-type       |     pid | status
---------------------------------+--------------------+---------+---------
webcache1                        | WebCache-admin     |   19556 | Alive
webcache1                        | WebCache           |   19555 | Alive
ohs1                             | OHS                |    7249 | Alive
```

To view the status of a particular component or component type, use the following command:

```
opmnctl status componentName [, componentType] [, componentSet
```

For example, to view the status of an Oracle Virtual Directory instance named ovd1, use the following command:

```
opmnctl status ias-component=ovd1
```

You can use OPMN to start and stop system components, monitor system components, and perform many other tasks related to process management. For example, you can use the following commands to start and stop OPMN and all OPMN-managed processes, such as Oracle HTTP Server and Oracle Web Cache:

```
opmnctl startall
opmnctl stopall
```

To start a component, use the following command:

```
opmnctl startproc componentName [, componentType] [, componentSet
```

For example, to start an Oracle HTTP Server instance named ohs1, use the following command:

```
opmnctl startproc ias-component=ohs1
```

See Also:

- Chapter 4 for information about starting and stopping your Oracle Fusion Middleware environment
- Chapter 10 for more information about monitoring your Oracle Fusion Middleware environment
- *Oracle Fusion Middleware Oracle Process Manager and Notification Server Administrator's Guide*

3.6 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.

See Also: "Understanding WebLogic Server MBeans" in the *Oracle Fusion Middleware Developing Custom Management Utilities With JMX for Oracle WebLogic Server*

Fusion Middleware Control provides a set of MBean browsers that allow to you 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 following topics describe how to view the MBeans:

- Using the System MBean Browser
- Using the MBeans for a Selected Application

3.6.1 Using the System MBean Browser

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 farm, then **WebLogic Domain,** and then 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.
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.**

See Also: The Fusion Middleware Control online help

3.6.2 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 3.6.1, 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 the farm, then **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.**

3.7 Managing Components

Oracle Fusion Middleware components include Oracle WebLogic Server, Java components that are part of Oracle SOA Suite and WebCenter, such as Oracle BPEL Process Manager or Oracle Business Activity Monitoring, and system components such as Oracle Web Cache.

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 OPMN, WLST, or Fusion Middleware Control.

See:

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

3.8 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.

> **See Also:** "Understanding Users and Roles" in the *Oracle Fusion Middleware Security Guide*:

You can change the password of the administrative user using the Oracle WebLogic Server Administration Console or the WLST command line.

3.8.1 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 *Oracle Fusion Middleware Oracle WebLogic Server MBean Reference*.

3.8.2 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 target navigation 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.**

3.9 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. See *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard*.
2. Configure components, such as Oracle SOA Suite, Oracle HTTP Server, or Oracle Web Cache. See *Oracle Fusion Middleware Installation Planning Guide*.

3. Configure SSL. See Chapter 6.
4. Deploy an application. See Chapter 9.
5. Configure load balancing. You can configure load balancing between different components or applications. See the *Oracle Fusion Middleware High Availability Guide*.
6. Backup your environment. See Chapter 15.
7. Monitor your environment and manage log files. See Chapter 10 and Chapter 11.

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

4

Starting and Stopping Oracle Fusion Middleware

This chapter describes procedures for starting and stopping Oracle Fusion Middleware.

It contains the following 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 or OPMN 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 Instances

You can start Oracle WebLogic Server Administration Servers using the WLST command line. You can start and stop Managed Servers using 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.

4.2.1 Starting and Stopping Administration Servers

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

To start an Administration Server, use the following script, which is based on WLST commands:

```
MW_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh
         -Dweblogic.management.username=weblogic
         -Dweblogic.management.password=password
         -Dweblogic.system.StoreBootIdentity=true
```

To stop an Administration Server, use the following script, which is based on WLST commands:

```
MW_HOME/user_projects/domains/domain_name/bin/stopWeblogic.sh
       username password admin_url
```

4.2.2 Starting and Stopping Managed Servers Using WLST

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

To start a WebLogic Server Managed Server, use the following script, which is based on WLST commands:

```
(UNIX) MW_HOME/user_projects/domains/domain_name/bin/startManagedWebLogic.sh
           managed_server_name admin_url
(Windows) MW_HOME\user_projects\domains\domain_name\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, which is based on WLST commands:

```
(UNIX) MW_HOME/user_projects/domains/domain_name/bin/stopManagedWebLogic.sh
            managed_server_name admin_url username password
(Windows) MW_HOME\user_projects\domains\domain_name\bin\stopManagedWebLogic.cmd
            managed_server_name admin_url username password
```

4.2.3 Starting and Stopping Managed Servers Using Fusion Middleware Control

Fusion Middleware Control, as well as 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 SOA Suite, Oracle WebCenter, or Oracle JRF, you must first configure Node Manager, as described in Section 4.2.4.

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

1. From the navigation pane, expand the farm, then **WebLogic Domain,** and then 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 on the server, then choose **Control,** then **Start Up** or **Shut Down.**

4.2.4 Configuring Node Manager to Start Managed Servers

If a Managed Server contains other Oracle Fusion Middleware products, such as Oracle SOA Suite, Oracle WebCenter, or Oracle JRF, the Managed Servers environment must be configured to set the correct classpath and parameters. This environment information is provided through the start scripts, such as startWebLogic and setDomainEnv, that are located in the domain directory.

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
MW_HOME/wl_server_n/server/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.

> **See Also:** "Using Node Manager" in the *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server* for other methods of configuring and starting Node Manager

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 sections describe how to start and stop components using the command line and Fusion Middleware Control.

4.3.1 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 use `opmnctl` commands to start and stop the 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 Directory Integration Platform, use the following command:

  ```
  startApplication("DIP")
  ```

- To start and stop system components, use the `opmnctl` command-line tool. It is located in the following directory:

  ```
  (UNIX) ORACLE_HOME/opmn/bin
  (Windows) ORACLE_HOME\opmn\bin
  ```

  To start or stop OPMN and all system processes, such as Oracle HTTP Server:

  ```
  opmnctl startall
  opmnctl stopall
  ```

  To start, stop, or restart a component using opmnctl:

  ```
  opmnctl startproc ias-component=component_name
  opmnctl stopproc ias-component=component_name
  opmnctl restartproc ias-component=component_name
  ```

  For example, to start Oracle HTTP Server, ohs1:

  ```
  opmnctl startproc ias-component=ohs1
  ```

  To start, stop, or restart the subprocess of a component:

  ```
  opmnctl stopproc process-type=process
  opmnctl startproc process-type=process
  opmnctl restartproc process-type=process
  ```

4.3.2 Starting and Stopping Components Using Fusion Middleware Control

To start, stop, and restart components:

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

Note: If OPMN is not started, you cannot start system components such as Oracle HTTP Server or Oracle Internet Directory using Fusion Middleware Control. To start OPMN, use the following command:

```
opmnctl start
```

4.4 Starting and Stopping Fusion Middleware Control

If Fusion Middleware Control is configured for a domain, it is automatically started when you start an Oracle WebLogic Server Administration Server, as described in Section 4.2.1.

If Fusion Middleware Control is configured for a domain, it is automatically stopped when you stop an Oracle WebLogic Server Administration Server, as described in Section 4.2.1.

4.5 Starting and Stopping Oracle Management Agent

Oracle Management Agent is designed specifically for monitoring particular Oracle Fusion Middleware components.

To start Oracle Management Agent, use the following command:

```
opmnctl startproc ias-component=EMAGENT
```

To stop Oracle Management Agent, use the following command:

```
opmnctl stopproc ias-component=EMAGENT
```

4.6 Starting and Stopping Applications

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

4.6.1 Starting and Stopping Java EE Applications Using WLST

To start or stop applications 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 the *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.

4.6.2 Starting and Stopping 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.**

To start or stop a SOA Composite application using Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **SOA,** and then **soa-infra.**
2. Select the application.
3. On the SOA Composite page, click **Start Up** or **Shut Down.**

4.7 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 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 and it is important to start and stop them in the proper order.

You can follow these procedures when you need to completely shut down your Oracle Fusion Middleware environment. For example, when preparing to perform a complete backup of your environment, or apply a patch.

4.7.1 Starting an Oracle Fusion Middleware Environment

To start an Oracle Fusion Middleware environment:

1. Start any database-based repository:
   1. Set the ORACLE_HOME environment variable to the Oracle home for the database.
   2. Set the ORACLE_SID environment variable to the SID for the database (default is `orcl`).
   3. Start the Net Listener:

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

   4. Start the database instance:

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

2. Start Oracle Identity Management system components:
   1. Set the ORACLE_HOME environment variable to the Oracle home for the Identity Management components.
   2. Start OPMN and all system components:

      ```
      opmnctl startall
      ```

3. Start the Oracle WebLogic Server Administration Server as described in Section 4.2.1.
4. Start the Oracle WebLogic Server Managed Servers as described in Section 4.2.2. Any applications deployed to the server are also started.
5. Start OPMN and all system components.
   1. Set the ORACLE_HOME and ORACLE_INSTANCE environment variables to the Oracle home and Oracle instance for the system components.
   2. Start OPMN and all system components in that Oracle instance:

      ```
      opmnctl startall
      ```

6. If your environment includes components that are targets monitored by Oracle Management Agent, start Oracle Management Agent, as described in Section 4.5.

4.7.2 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.
   1. Set the ORACLE_HOME and ORACLE_INSTANCE environment variables to the Oracle home and Oracle instance for the system components.
   2. Stop OPMN and all system components in that Oracle instance:

      ```
      opmnctl stopall
      ```

2. If your environment includes components that are targets monitored by Oracle Management Agent, stop Oracle Management Agent, as described in Section 4.5.

3. Stop the Oracle WebLogic Server Managed Servers as described in Section 4.2. Any applications deployed to the server are also stopped.
4. Stop Oracle Identity Management components:
   a. Set the ORACLE_HOME environment variable to the Oracle home for the Identity Management components.
   b. Stop OPMN and all system components:

   ```
   opmnctl stopall
   ```

5. Stop the Administration Server as described in Section 4.2.1.
6. Stop any database-based repository:
   a. Set the ORACLE_HOME environment variable to the Oracle home for the database.
   b. Set the ORACLE_SID environment variable to the SID for the database (default is `orcl`).
   c. Stop the database instance:

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

   d. Stop the Net Listener:

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

4.8 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

4.8.1 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:** *Oracle Fusion Middleware High Availability Guide* for information about starting and stopping in high-availability environments

4.8.2 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.

> **See Also:** *Oracle Database Administrator's Guide* in the Oracle Database 11*g* documentation library

5 Managing Ports

This chapter describes how to view and change Oracle Fusion Middleware port numbers.

It contains the following topics:

- 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 installation. 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 installation.

> **See Also:** Appendix C for a complete list of allotted port ranges. 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 current port numbers for system components, use the following command:

```
(UNIX) ORACLE_INSTANCE/bin/opmnctl status -l
(Windows) ORACLE_INSTANCE\bin\opmnctl status -l
```

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

```
wls:/SOA_domain/serverConfig> get('AdministrationPort')
```

```
9002
```

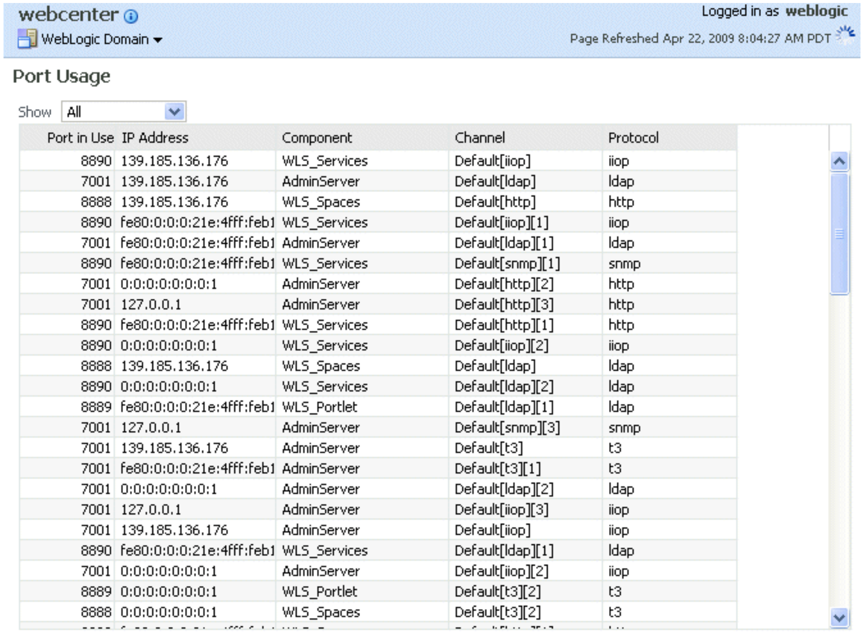
5.2.2 Viewing Port Numbers Using Fusion Middleware Control

You can view the port numbers of the domain, the Administration Server, Managed Servers, or components, such as the SOA Infrastructure and Oracle Web Cache, using Fusion Middleware Control.

For example, to view the ports of a domain:

1. From the navigation pane, expand the farm and then **WebLogic Domain.**
2. Select the domain.
3. From the WebLogic Domain menu, choose **Port Usage.**

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



Port in Use	IP Address	Component	Channel	Protocol
8890	139.185.136.176	WLS_Services	Default[iiop]	iiop
7001	139.185.136.176	AdminServer	Default[ldap]	ldap
8888	139.185.136.176	WLS_Spaces	Default[http]	http
8890	fe80:0:0:0:21e:4fff:feb1	WLS_Services	Default[iiop][1]	iiop
7001	fe80:0:0:0:21e:4fff:feb1	AdminServer	Default[ldap][1]	ldap
8890	fe80:0:0:0:21e:4fff:feb1	WLS_Services	Default[snmp][1]	snmp
7001	0:0:0:0:0:0:0:1	AdminServer	Default[http][2]	http
7001	127.0.0.1	AdminServer	Default[http][3]	http
8890	fe80:0:0:0:21e:4fff:feb1	WLS_Services	Default[http][1]	http
8890	0:0:0:0:0:0:0:1	WLS_Services	Default[iiop][2]	iiop
8888	139.185.136.176	WLS_Spaces	Default[ldap]	ldap
8890	0:0:0:0:0:0:0:1	WLS_Services	Default[ldap][2]	ldap
8889	fe80:0:0:0:21e:4fff:feb1	WLS_Portlet	Default[ldap][1]	ldap
7001	127.0.0.1	AdminServer	Default[snmp][3]	snmp
7001	139.185.136.176	AdminServer	Default[t3]	t3
7001	fe80:0:0:0:21e:4fff:feb1	AdminServer	Default[t3][1]	t3
7001	0:0:0:0:0:0:0:1	AdminServer	Default[ldap][2]	ldap
7001	127.0.0.1	AdminServer	Default[iiop][3]	iiop
7001	139.185.136.176	AdminServer	Default[iiop]	iiop
8890	fe80:0:0:0:21e:4fff:feb1	WLS_Services	Default[ldap][1]	ldap
7001	0:0:0:0:0:0:0:1	AdminServer	Default[iiop][2]	iiop
8889	0:0:0:0:0:0:0:1	WLS_Portlet	Default[t3][2]	t3
8888	0:0:0:0:0:0:0:1	WLS_Spaces	Default[t3][2]	t3

   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 the SOA Infrastructure and Oracle Web Cache, 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, as long as it is an unused port. You do not have to use a port in the allotted port range for the component. See Appendix C 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 Oracle Web Cache Ports
- Changing OPMN Ports (ONS Local, Request, and Remote)
- Changing Oracle Portal Ports
- Changing the Oracle Database Net Listener Port

For information about changing Oracle Internet Directory ports, see "Configuring Server Properties" or "Setting System Configuration Attributes by Using ldapmodify" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

5.3.1 Changing the Oracle WebLogic Server Listen Ports

You can change the HTTP listen port and the HTTPS (SSL) 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 Fusion Middleware Control
- Changing the Oracle WebLogic Server Listen Ports Using WLST

See Also: *Oracle Fusion Middleware Configuring 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 Fusion Middleware Control

To change the HTTP listen port and the HTTPS (SSL) 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 server setting page is displayed.

2. 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. For example for Oracle Portal, if the listen port for the Oracle WebLogic Server configured as WLS_PORTAL is changed, then you must make a corresponding change to the configuration in Oracle HTTP Server, which is pointing to the older port. Change the port number in the following file:

   ```
   ORACLE_INSTANCE/OHS/ohs_name/moduleconf/portal.conf
   ```

5.3.1.2 Changing the Oracle WebLogic Server Listen Ports Using WLST

To change the HTTP listen port and the HTTPS (SSL) 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("MW_HOME/user_projects/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. For example, if you are using Oracle Web Cache to improve the performance of your Oracle Fusion Middleware environment, you must modify the Oracle Web Cache origin server settings whenever you modify the Oracle HTTP Server Listen ports.

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
- Changing the Oracle HTTP Server SSL Listen Port

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

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.

- Task 1, "Modify the Oracle HTTP Server Listen Port"
- Task 2, "Update Oracle Web Cache"
- Task 3, "Restart the System Components"

Task 1 Modify the Oracle HTTP Server Listen Port

To change the Oracle HTTP Server Listen port:

1. From the navigation pane, expand the farm, then **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.**)

Task 2 Update Oracle Web Cache

If you are using Oracle Web Cache as a reverse proxy, you must update Oracle Web Cache:

1. From the Fusion Middleware Control navigation pane, expand the farm, then **Web Tier.** Select the Oracle Web Cache instance.
2. From the Web Cache menu, choose **Administration,** then **Origin Servers.**
3. Select the origin server for which you have changed the port, and click **Edit.**

   The Edit Origin Server page is displayed.
4. In the **Port** field, change the port number.
5. Click **OK.**
6. Restart Oracle Web Cache. (From the Web Cache menu, choose **Control,** then **Restart.**)

Task 3 Restart the System Components

Restart OPMN and all system components in that Oracle instance:

```
opmnctl stopall
opmnctl startall
```

5.3.2.3 Changing the Oracle HTTP Server SSL Listen Port

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.

- Task 1, "Modify the Oracle HTTP Server SSL Listen Port"
- Task 2, "Update Oracle Web Cache"
- Task 3, "Re-register mod_osso"
- Task 4, "Restart System Components"

Task 1 Modify the Oracle HTTP Server SSL Listen Port

To change the Oracle HTTP Server SSL Listen port:

1. From the navigation pane, expand the farm, then **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.**)

Task 2 Update Oracle Web Cache

If you are using Oracle Web Cache as a reverse proxy, you must update Oracle Web Cache:

1. From the Fusion Middleware Control navigation pane, expand the farm, then **Web Tier.** Select the Oracle Web Cache instance.
2. From the Web Cache menu, choose **Administration,** then **Origin Servers.**
3. Select the origin server for which you have changed the port, and click **Edit.**

   The Edit Origin Server page is displayed.
4. In the **Port** field, change the port number.
5. Click **OK.**
6. Restart Oracle Web Cache. (From the Web Cache menu, choose **Control,** then **Restart.**)

Task 3 Re-register mod_osso

If you are using Oracle Single Sign-On, you must use Release 10.1.4.3. If you have enabled Oracle Single Sign-On authentication (that is, you registered mod_osso), follow these steps to re-register mod_osso:

1. On the Oracle Single Sign-On host, set the environment variables ORACLE_HOME and ORACLE_SID.
2. On the Oracle Single Sign-On host, run the `ssoreg` script, using the `-remote_midtier` option. The script is located at:

   ```
   (UNIX) ORACLE_HOME/sso/bin/ssoreg.sh
   (Windows)ORACLE_HOME\sso\bin\ssoreg.bat
   ```

   For example, on LINUX:

   ```
   $ORACLE_HOME/sso/bin/ssoreg.sh -oracle_home_path $ORACLE_HOME
    -config_mod_osso TRUE
    -site_name example.com:7778
    -remote_midtier
    -config_file $ORACLE_HOME/Apache/Apache/conf/osso/myosso.conf
    -mod_osso_url http://example.com:7778
   ```

   The resulting configuration file (`myosso.conf` in the example) is an obfuscated osso configuration file.
3. Copy the obfuscated osso configuration file to the Oracle HTTP Server host moduleconf directory for editing:

   ```
   ORACLE_INSTANCE/config/OHS/ohs_name/moduleconf
   ```

Task 4 Restart System Components

Restart OPMN and the system components in that Oracle instance:

```
opmnctl stopall
opmnctl startall
```

5.3.3 Changing Oracle Web Cache Ports

You can change the HTTP and HTTPS listen ports, the administration port, the statistics port and the invalidation port for Oracle Web Cache using Fusion Middleware Control.

To change the port number:

1. From the navigation pane, expand the farm, then **Web Tier,** then select the Oracle Web Cache instance.
2. From the Web Cache menu, choose **Administration,** then **Ports Configuration.**
3. Select a port, then click **Edit.**
4. Change the port number, then click **OK.**
5. Restart Oracle Web Cache. (From the Web Cache menu, choose **Control,** then **Restart.**)
6. If you reconfigure the Web Cache invalidation port and you use Oracle Portal, you must update the port information maintained by Oracle Portal, as described in Section 5.3.5.2.

5.3.4 Changing OPMN Ports (ONS Local, Request, and Remote)

This section describes how to change any of the following port numbers:

- ONS Local port
- ONS Request port
- ONS Remote port

To change these ports:

1. Stop OPMN, and all OPMN-managed processes:

   ```
   (UNIX) ORACLE_INSTANCE/bin/opmnctl stopall
   (Windows)ORACLE_INSTANCE\bin\opmnctl stopall
   ```

2. Open the `opmn.xml` file:

   ```
   (UNIX) ORACLE_INSTANCE/config/OPMN/opmn
   (Windows) ORACLE_INSTANCE\config\OPMN\opmn
   ```

3. Under the `<notification-server>` element, modify the `local`, `remote`, or `request` parameter, depending on the port you are changing, in the `<port>` element, and then save the file.

   For example:

   ```
   <port local="6101" remote="6201" request="6004"/>
   ```

4. Start OPMN, and all OPMN-managed processes:

   ```
   (UNIX) ORACLE_INSTANCE/bin/opmnctl startall
   (Windows) ORACLE_INSTANCE\bin\opmnctl startall
   ```

5.3.5 Changing Oracle Portal Ports

Oracle Portal maintains information about some of the ports used by its underlying components. This section describes how to manage Oracle Portal ports. It includes the following topics:

- Changing the Oracle Portal Midtier Port

- Changing the Oracle Web Cache Invalidation Port for Oracle Portal
- Changing Oracle Portal Oracle Internet Directory Port
- Changing PPE Loopback Port
- Changing Oracle Portal SQL*Net Listener Port
- Restarting WLS_PORTAL Managed Server

> **Note:** When you change these ports as described in this section, only the Oracle Portal configuration is updated. To update or change the port numbers of an underlying component, such as Oracle Web Cache or Oracle Internet Directory, see the component-specific documentation for information about managing ports.
>
> The configuration procedures described in this section require you to restart the WLS_PORTAL Managed Server.

5.3.5.1 Changing the Oracle Portal Midtier Port

In a default installation, you can access Oracle Portal through the Oracle Web Cache port, such as 8090. This port is referred to as the Oracle Portal midtier port. You must update this port if Oracle Web Cache is configured to listen on a different port or Oracle Web Cache is front-ended by a Proxy or Load Balancing Router (LBR).

To change the Oracle Portal midtier port in Fusion Middleware Control:

1. From the navigation pane, expand the farm, choose **Portal**, and select the Portal instance.
2. From the Portal menu, choose **Settings**, and then **Wire Configuration**.
3. Select the Database Access Descriptor, such as `portal`.
4. Expand the Portal Midtier section.
5. Change the port number, and click **Apply**.
6. Restart the WLS_PORTAL Managed Server. For more information, see Section 5.3.5.6.

5.3.5.2 Changing the Oracle Web Cache Invalidation Port for Oracle Portal

Oracle Portal caches content in Oracle Web Cache. When content changes, Oracle Portal invalidates such cached content and maintains the Oracle Web Cache invalidation port. If you reconfigure the Web Cache invalidation port, you must update the port information maintained by Oracle Portal.

To change the Oracle Portal Invalidation port in Fusion Middleware Control:

1. From the navigation pane, expand the farm, choose **Portal**, and select the Portal instance.
2. From the Portal menu, choose **Settings**, and then **Wire Configuration**.
3. Select the Database Access Descriptor, such as `portal`.
4. Expand the Web Cache section.
5. Change the Invalidation Port number. If the Invalidation user name and the password are blank, enter the user name and the password.

> **Note:** The Port number, Invalidation user name, and Invalidation password entered here must match the corresponding values of the Oracle Web Cache instance used by Oracle Portal. For more information about resetting these values, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache*.

6. Click **Apply**.
7. Restart the WLS_PORTAL Managed Server. For more information, see Section 5.3.5.6.

5.3.5.3 Changing Oracle Portal Oracle Internet Directory Port

Oracle Portal maintains information about Oracle Internet Directory ports.

To change the Oracle Portal Oracle Internet Directory (OID) port in Fusion Middleware Control:

1. From the navigation pane, expand the farm, choose **Portal**, and select the Portal instance.
2. From the Portal menu, choose **Settings**, and then **Wire Configuration**.
3. Select the Database Access Descriptor, such as `portal`.
4. Expand the OID section.
5. Change the port number.
6. Enter the Oracle Internet Directory user name and the password.
7. Click **Apply**.
8. Restart the WLS_PORTAL Managed Server. For more information, see Section 5.3.5.6.

5.3.5.4 Changing PPE Loopback Port

While servicing Portal pages, Oracle Portal makes loopback calls using the default site port. In some configurations, such as external SSL, you must configure the loopback call to a port other than the default site port.

To change the PPE Loopback port in Fusion Middleware Control:

1. From the navigation pane, expand the farm, choose **Portal**, and select the Portal instance.
2. From the Portal menu, choose **Settings**, and then **Page Engine**.
3. Expand the Advanced Properties section.
4. Change the port number in the **Use Port**.
5. Specify the protocol in the **Use Protocol** field.
6. Click **Apply**.
7. Restart the WLS_PORTAL Managed Server. For more information, see Section 5.3.5.6.

5.3.5.5 Changing Oracle Portal SQL*Net Listener Port

Oracle Portal maintains information about the repository connection in the `host:port:servicename` format inside a Database Access Descriptor (in a file named `portal_dads.conf`). If the SQL*Net listener is reconfigured to listen on a different port, you must reconfigure this port value in Oracle Portal.

To change the Oracle Portal SQL*Net Listener port in Fusion Middleware Control:

1. From the navigation pane, expand the farm, choose **Portal**, and select the Portal instance.
2. From the Portal menu, choose **Settings**, and then **Database Access Descriptor**.
3. Select the Database Access Descriptor, such as `/pls/portal`.
4. Click **Edit**.
5. Expand the Portal Database Access Details section.
6. Update the **Database Connect String** field to reflect the new port.
7. Click **OK**.
8. Restart the WLS_PORTAL Managed Server. For more information, see Section 5.3.5.6.

5.3.5.6 Restarting WLS_PORTAL Managed Server

To restart WLS_PORTAL Managed Server in Fusion Middleware Control:

1. Expand the Farm domain, such as `Farm_ClassicDomain`.
2. Expand WebLogic Domain.
3. Expand the domain, such as `Classic Domain`.
4. Expand cluster_portal, when applicable.
5. Choose WLS_PORTAL.
6. From the WLS_PORTAL WebLogic Server menu, choose **Control**, then **Shut Down**. Ensure that the status of WLS_PORTAL shows Down.
7. From the WLS_PORTAL WebLogic Server menu, choose **Control**, then **Start Up**. Ensure that the status of WLS_PORTAL shows Up.

5.3.6 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, it is possible that both databases can use the same port.

Note that multiple Oracle Database 10*g* and Oracle Database 11*g* 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 10*g* or Oracle Database 11*g* database, they can all use port 1521. There is no need to change the listener port number.

Note: If you want to run two listeners that use the same key value on one host, refer to Section 5.3.6.1, "Changing the KEY Value for an IPC Listener"

A metadata repository may be used in several different ways. Use the following table to determine the steps that are required for changing your type of metadata repository:

If the Metadata Repository is used as follows:	Follow these tasks to change its Oracle Net listener port:
Identity Management repository and product metadata repository	Task 1, "Stop Components" Task 2, "Change the Metadata Repository for Oracle Net Listener Port" Task 3, "Change the System Data Source" Task 4, "Update Oracle Internet Directory" Task 5, "Update Oracle Single Sign-On" Task 6, "Update Oracle Portal" Task 7, "Update Other Components"
Identity Management repository only	Task 1, "Stop Components" Task 2, "Change the Metadata Repository for Oracle Net Listener Port" Task 4, "Update Oracle Internet Directory" Task 5, "Update Oracle Single Sign-On"
Product metadata repository	Task 1, "Stop Components" Task 2, "Change the Metadata Repository for Oracle Net Listener Port" Task 3, "Change the System Data Source" Task 4, "Update Oracle Internet Directory" Task 6, "Update Oracle Portal" Task 7, "Update Other Components"

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 4, "Update Oracle Internet Directory"
- Task 5, "Update Oracle Single Sign-On"
- Task 7, "Update Other Components"

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 like 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
   ```

8. Start Oracle Internet Directory:

   ```
   opmnctl start
   opmnctl startproc ias-component=OID
   ```

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 use Oracle WebLogic Server Administration Console:

1. In the Change Center, click **Lock & Edit.**

2. In the Domain Structure section, expand **Services,** then **JDBC**, and select **Data Sources.**

   The Summary of JDBC Data Sources page is displayed.

3. Select the data source you want to change.

   The Settings page is displayed.

4. Select the Connection Pool tab.

5. To change the database port, modify the **URL** field. For example:

   ```
   jdbc:oracle:thin:@hostname.domainname.com:1522/orcl
   ```

6. Click **Save.**

7. Restart the servers that use this data source. (Click the Target tab to see the servers that use this data source.)

Task 4 Update Oracle Internet Directory

On the Identity Management host, update Oracle Internet Directory with the new Oracle Net listener port number:

1. Update the port number in tnsnames.ora file, which is located in the following directory:

   ```
   (UNIX) ORACLE_INSTANCE/config
   (Windows) ORACLE_INSTANCE\config
   ```

2. Update the registration of the component with the Administration Server, using the opmnctl `updatecomponentregistration` command with the new port number, as shown in the following example:

   ```
   opmnctl updatecomponentregistration -Db_info DBHostName:TNSPORT:DBSERVICENAME
      -componentName oid1 -componentType OID
   ```

3. Start OPMN and all processes in the Oracle instance in the Oracle Internet Directory Oracle home:

   ```
   opmnctl startall
   ```

Task 5 Update Oracle Single Sign-On

If you are using Oracle Single Sign-On, from the Oracle Single Sign-On Oracle home:

1. On UNIX systems, set the LD_LIBRARY_PATH, LD_LIBRARY_PATH_64, LIB_PATH, or SHLIB_PATH environment variables to the proper values, as shown in Table 3–1. The actual environment variables and values that you must set depend on the type of your UNIX operating system.

2. Update Oracle Single Sign-On with the new repository port number by executing the following command:

   - On UNIX systems:

     ```
     $ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/sso/lib/ossoca.jar reassoc
     -repos $ORACLE_HOME
     ```

   - On Windows systems:

     ```
     %ORACLE_HOME%\jdk\bin\java -jar %ORACLE_HOME%\sso\lib\ossoca.jar reassoc
     -repos %ORACLE_HOME%
     ```

Task 6 Update Oracle Portal

To update Oracle Portal, follow the steps in Section 5.3.5.5.

Task 7 Update Other Components

In each Oracle instance that uses the metadata repository:

1. Update the following file with the new Oracle Net listener port number:

   ```
   (UNIX) ORACLE_INSTANCE/config/tnsnames.ora
   (Windows) ORACLE_INSTANCE\config\tnsnames.ora
   ```

2. Check the following file:

   ```
   (UNIX) ORACLE_HOME/ohs/conf/dads.conf
   (Windows) ORACLE_HOME\ohs\modplsql\conf\dads.conf
   ```

   Locate the line that begins with `PlsqlDatabaseConnectString`.

- If the line ends with `ServiceNameFormat` or `SIDFormat`, update the line with the new MDS Repository port number, save the file, and restart Oracle HTTP Server.
- If the line ends with `NetServiceNameFormat`, you do not need to do anything.

3. Start the components that use the metadata repository, as described in Section 4.3.

5.3.6.1 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, "SSL Configuration in Oracle Fusion Middleware"
- Chapter 7, "Managing Keystores, Wallets, and Certificates"

6 SSL Configuration in Oracle Fusion Middleware

You can configure Oracle Fusion Middleware to secure communications between Oracle Fusion Middleware components using SSL, which is an industry standard for securing communications. Oracle Fusion Middleware supports SSL version 3, as well as TLS version 1.

> **Note:** SSL version 2 has been de-supported in 11*g* Release 1 (11.1.1) due to security concerns; components or applications that used SSL version 2 in pre-11*g* Release 1 (11.1.1) will automatically be upgraded to use other SSL versions, that is, SSL version 3 and 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, then 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

Private, or 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. This 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 messages encrypted 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. The private key is securely stored, together with other security credentials, in an encrypted container such as an Oracle wallet.

Public key algorithms can guarantee the secrecy of a message, but 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, a process called authentication. Authentication can be accomplished through a certificate authority (CA), which is a third party trusted by both of the communicating parties.

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, Oracle Virtual Directory uses a JKS keystore to store keys and certificates. Configuring SSL for Oracle Virtual Directory thus requires setting up and using JKS keystores.

Other 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.

Components that use Oracle wallet include:

- Oracle HTTP Server
- Oracle Web Cache
- Oracle Internet Directory

Configuring SSL for these components thus requires setting up and using Oracle wallets.

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 using the server's certificate.

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

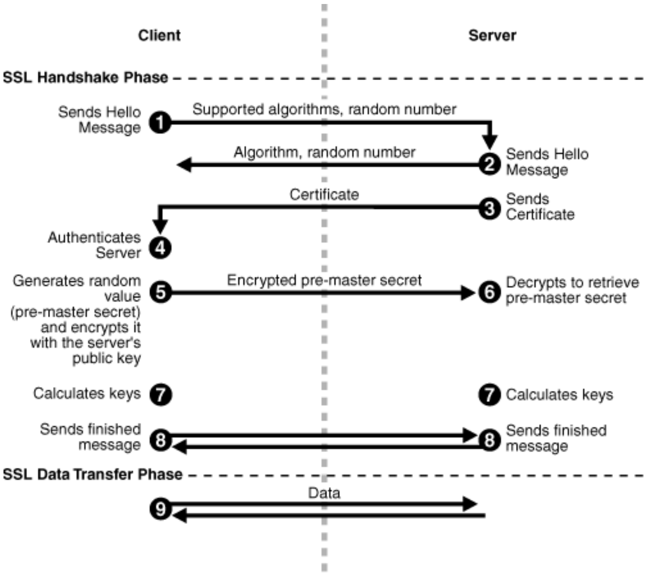
   The encryption keys are symmetric, that is, the same key is used to encrypt and decrypt the data.

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



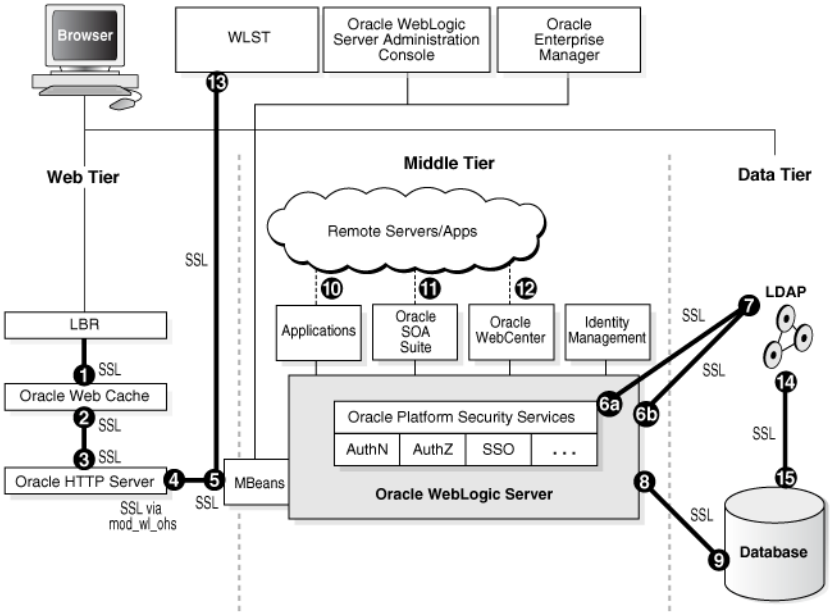
6.2 About SSL in Oracle Fusion Middleware

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 Oracle Enterprise Manager Fusion Middleware Control 11*g* user interface.
- Other administrative tools, such as `opmn`, 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 Inbound SSL for Oracle Web Cache Using Fusion Middleware Control" and Section 6.4.2.2, "Enable Inbound SSL for Oracle Web Cache Using WLST"
2. Section 6.4.2.3, "Enable Outbound SSL for Oracle Web Cache Using Fusion Middleware Control" and Section 6.4.2.4, "Specify the Wallet for Outbound SSL from Oracle Web Cache Using WLST"
3. Section 6.4.3.1, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control" and Section 6.4.3.2, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using WLST"
4. Section 6.4.3.3, "Enable SSL for Outbound Requests from Oracle HTTP Server"
5. Section 6.5.1.1, "Inbound SSL to Oracle WebLogic Server"
6. 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 6.5.1.2.2, "Outbound SSL from LDAP Authenticator to LDAP"
7. Section 6.6.1.1, "Enable Inbound SSL on an Oracle Internet Directory Listener Using Fusion Middleware Control" and Section 6.6.1.2, "Enabling Inbound SSL on an Oracle Internet Directory Listener Using WLST"
8. Section 6.6.3.2, "SSL-Enable a Data Source"
9. Section 6.6.3.1, "SSL-Enable Oracle Database"
10. Section 6.5.6, "Client-Side SSL for Applications"
11. Section 6.5.2, "Configuring SSL for Oracle SOA Suite"
12. Section 6.5.3, "Configuring SSL for Oracle WebCenter"
13. Section 6.3.3, "WLST Command-Line Tool"
14. Section 6.6.1.3, "Enabling Outbound SSL from Oracle Internet Directory to Oracle Database"
15. Section 6.6.3.1, "SSL-Enable Oracle Database"

In addition, you can configure SSL for identity management components. For details, see:

- Section 6.5.4.1, "Configuring SSL for Oracle Directory Integration Platform"
- Section 6.5.4.2, "Configuring SSL for Oracle Identity Federation"
- Section 6.5.4.3, "Configuring SSL for Oracle Directory Services Manager"

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 supports two types of keystores for keys and certificates:

- JKS-based keystore and truststore
- Oracle wallet

In 11*g* Release 1 (11.1.1), all Java components and applications use the JKS keystore. Thus all Java components and applications running on Oracle WebLogic Server use the JKS-based KeyStore and TrustStore.

The following system components continue to use the Oracle wallet:

- Oracle HTTP Server
- Oracle Web Cache
- Oracle Internet Directory

You can use Fusion Middleware Control or the command-line WLST and `orapki` interfaces, to manage wallets and their certificates for these system components. You can use either the Fusion Middleware Control or WLST to SSL-enable the listeners for these components.

Oracle Virtual Directory uses a JKS-based keystore. You can use Fusion Middleware Control or WLST to manage JKS keystores and their certificates for Oracle Virtual Directory. You can use either the Fusion Middleware Control or WLST to SSL-enable the listeners for Oracle Virtual Directory.

JDK's `keytool` utility manages the keystore used by Oracle WebLogic Server listeners for Java EE applications. This is the only keystore tool to manage these keystores; no GUI tool is available for this purpose.

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.

- 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.

Advanced Tools

- Oracle Wallet Manager GUI interface
- `orapki` command-line interface

These tools allow 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.

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.

6.4 Configuring SSL for the Web Tier

This section contains these topics:

- Configuring Load Balancers
- Enabling SSL for Oracle Web Cache Endpoints
- Enabling SSL for Oracle HTTP Server Virtual Hosts

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 Web Cache Endpoints

This section explains how to enable SSL for Oracle Web Cache listening endpoints using Fusion Middleware Control and WLST.

6.4.2.1 Enable Inbound SSL for Oracle Web Cache Using Fusion Middleware Control

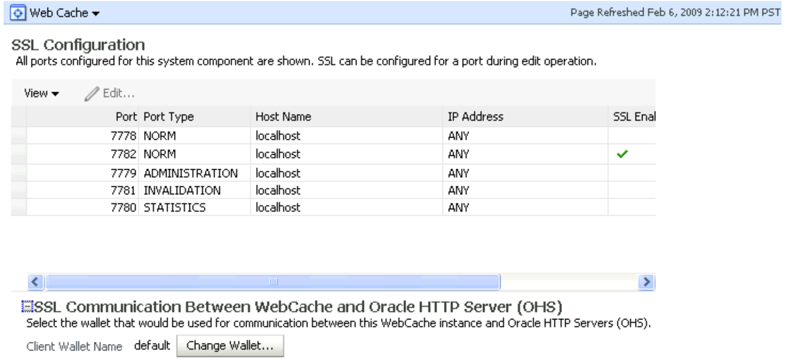
You can SSL-enable inbound traffic to Oracle Web Cache listening endpoints using these steps:

Note: This information applies only to inbound communication; for information about SSL-enabling outbound traffic from Oracle Web Cache to Oracle HTTP Server, see Section 6.4.2.3, "Enable Outbound SSL for Oracle Web Cache Using Fusion Middleware Control".

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

   For details about wallet creation and maintenance, see Chapter 7, "Managing Keystores, Wallets, and Certificates".
3. Navigate to **Oracle Web Cache**, then **Security**, then **SSL Configuration**.

The SSL Configuration page contains two sets of information:



Page Refreshed Feb 6, 2009 2:12:21 PM PST

SSL Configuration

All ports configured for this system component are shown. SSL can be configured for a port during edit operation.

Port	Port Type	Host Name	IP Address	SSL Enal
7778	NORM	localhost	ANY	
7782	NORM	localhost	ANY	✓
7779	ADMINISTRATION	localhost	ANY	
7781	INVALIDATION	localhost	ANY	
7780	STATISTICS	localhost	ANY	

SSL Communication Between WebCache and Oracle HTTP Server (OHS)

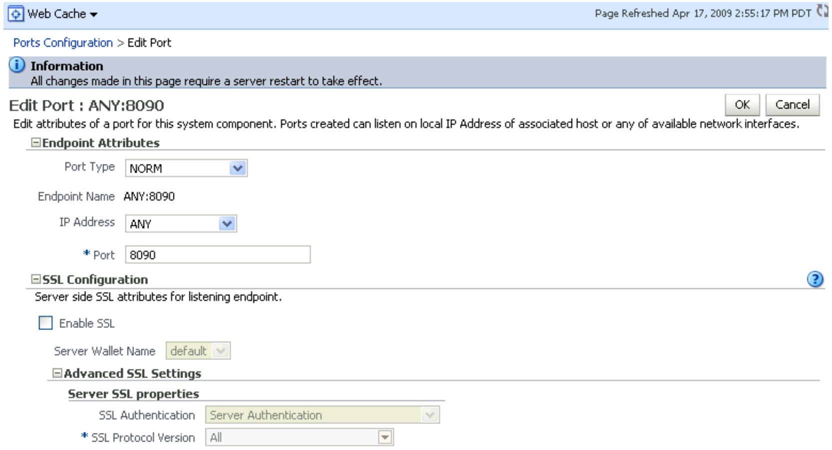
Select the wallet that would be used for communication between this WebCache instance and Oracle HTTP Servers (OHS).

Client Wallet Name default

The top table shows the inbound settings for a list of listening endpoints. A check in the **SSL Enabled** column indicates that the endpoint is configured for SSL.

The bottom portion of the page shows outbound SSL configuration. For more information about outbound SSL, see Section 6.4.2.3, "Enable Outbound SSL for Oracle Web Cache Using Fusion Middleware Control.".

4. Select an endpoint, and click **Edit**.



Page Refreshed Apr 17, 2009 2:55:17 PM PDT

Ports Configuration > Edit Port

Information

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

Edit Port : ANY:8090

Edit attributes of a port for this system component. Ports created can listen on local IP Address of associated host or any of available network interfaces.

Endpoint Attributes

Port Type NORM

Endpoint Name ANY:8090

IP Address ANY

* Port 8090

SSL Configuration

Server side SSL attributes for listening endpoint.

☐ Enable SSL

Server Wallet Name default

Advanced SSL Settings

Server SSL properties

SSL Authentication Server Authentication

* SSL Protocol Version All

The Edit Port page appears. This page contains two sections—a top portion with general details like port and IP address, and a bottom section that configures SSL parameters.

5. To disable SSL, uncheck **Enable SSL**; restart the component instance by navigating to Oracle Web Cache, then **Control**, then **Restart**.

6. To enable SSL for this endpoint, check **Enable SSL**. Next, enter SSL configuration parameters:

   - Select an Oracle wallet from the drop-down list.

   > **Note:** Ensure that the wallet contains the server certificate and its corresponding CA certificate.

   - Select the type of SSL authentication.
   - Select the protocol version (the available options are determined by your choice of authentication).

7. Click **OK**.
8. Restart the Oracle Web Cache instance by navigating to **Oracle Web Cache**, then **Control**, then **Restart.**

> **See Also:** Section 7.4.1.3, "Sharing Wallets Across Instances"

6.4.2.2 Enable Inbound SSL for Oracle Web Cache Using WLST

You can enable SSL for inbound traffic to Oracle Web Cache using the WLST command-line tool.

SSL-Enable Oracle Web Cache Inbound in server-auth Mode Using WLST

Take these steps:

1. Determine the listening endpoints for this Oracle Web Cache instance by running the following command:

   ```
   listListeners('inst1','wc1')
   ```

   This command will list all the listening endpoints for this instance; select the one that needs to be configured for SSL. For example, select `CACHE.index1.LISTEN.index1`.

2. Configure the listening endpoint with SSL properties:

   ```
   configureSSL('inst1',
      'wc1',
      'webcache',
      'CACHE.index1.LISTEN.index1')
   ```

> **Note:**
>
> - `configureSSL` uses defaults for all SSL attributes; see Table 6–5 for details.
> - You may also specify a properties file as a parameter to `configureSSL`; see Table 6–4 for details.

6.4.2.3 Enable Outbound SSL for Oracle Web Cache Using Fusion Middleware Control

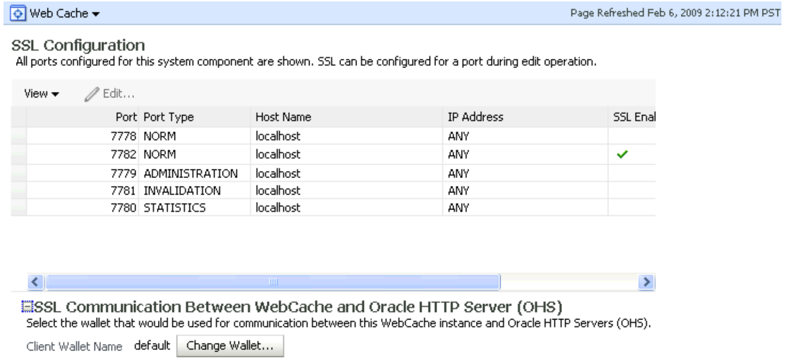
Outbound Oracle Web Cache refers to traffic from Oracle Web Cache to Oracle HTTP Server.

There are two aspects to set up SSL for outbound traffic from Oracle Web Cache: selecting a wallet for outbound SSL and configuring SSL.

Wallet Selection

Take these steps:

1. Navigate to **Oracle Web Cache**, then **Security**, then **SSL Configuration**.



SSL Configuration

All ports configured for this system component are shown. SSL can be configured for a port during edit operation.

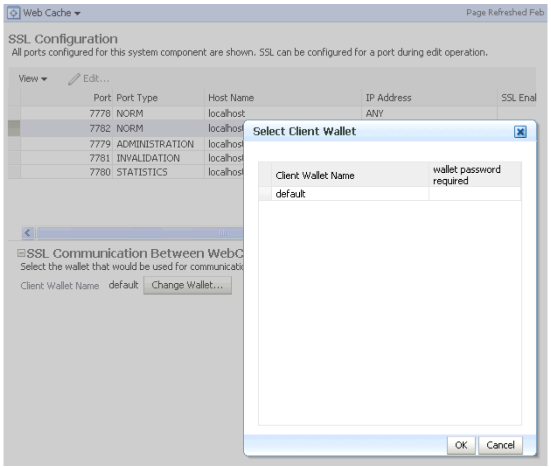
Port	Port Type	Host Name	IP Address	SSL Enal
7778	NORM	localhost	ANY	
7782	NORM	localhost	ANY	✓
7779	ADMINISTRATION	localhost	ANY	
7781	INVALIDATION	localhost	ANY	
7780	STATISTICS	localhost	ANY	

SSL Communication Between WebCache and Oracle HTTP Server (OHS)

Select the wallet that would be used for communication between this WebCache instance and Oracle HTTP Servers (OHS).

Client Wallet Name default

2. At the bottom of the page, click **Change Wallet** to display the available wallets for this listener.



Select Client Wallet

Client Wallet Name	wallet password required
default	

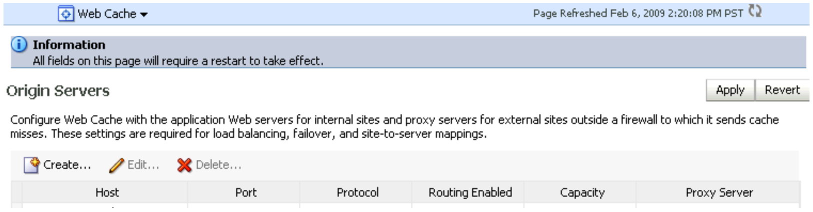
3. Select the desired wallet for outbound SSL and click **OK**.

SSL Configuration

Take these steps:

1. Navigate to the Oracle Web Cache instance, then **Administration**, then **Origin Servers**.

   This page displays the Oracle HTTP Servers with which this Oracle Web Cache instance can communicate. For example, if Oracle Web Cache can talk to two different Oracle HTTP Servers you would see two rows in the table.



Information

All fields on this page will require a restart to take effect.

Origin Servers

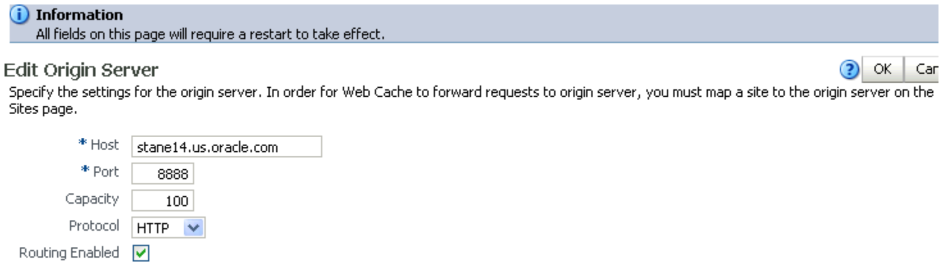
Configure Web Cache with the application Web servers for internal sites and proxy servers for external sites outside a firewall to which it sends cache misses. These settings are required for load balancing, failover, and site-to-server mappings.

Host	Port	Protocol	Routing Enabled	Capacity	Proxy Server

   In this example, the Oracle Web Cache instance is currently configured for non-SSL communication to the origin server over this host and port.

2. To enable SSL for outbound traffic to this origin server, select the row and click **Edit**.

3. The Edit Origin Server page appears:



Information
All fields on this page will require a restart to take effect.

Edit Origin Server

Specify the settings for the origin server. In order for Web Cache to forward requests to origin server, you must map a site to the origin server on the Sites page.

* Host stane14.us.oracle.com
* Port 8888
Capacity 100
Protocol HTTP
Routing Enabled ☑

4. Use the Protocol drop-down box to change the protocol to `https`.
5. Click **OK**. Oracle Web Cache is now configured to communicate to the origin server over SSL.

> **Note:** When editing the origin server settings on this page, ensure that Oracle HTTP Server is listening at this port in SSL mode.

6.4.2.4 Specify the Wallet for Outbound SSL from Oracle Web Cache Using WLST

To change the wallet in use for outbound SSL connections from Oracle Web Cache, use a command like the following:

```
configureSSL('inst1',
   'wc1',
   'webcache',
   'CACHE.index0.CLIENTSSL',
   'property-file.prop')
```

where:

- `inst1` is the name of the application server instance
- `wc1` is the name of the Oracle Web Cache instance
- `webcache` is the component type
- `CACHE.index0.CLIENTSSL` is the listener name for client SSL
- `property-file.prop` contains:

  `KeyStore=`*`wallet-path`*

6.4.3 Enabling SSL for Oracle HTTP Server Virtual Hosts

This section shows how to manage SSL configuration for Oracle HTTP Server virtual hosts.

For inbound traffic:

- Section 6.4.3.1, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control"
- Section 6.4.3.2, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using WLST"

For outbound traffic:

- Section 6.4.3.3, "Enable SSL for Outbound Requests from Oracle HTTP Server"

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

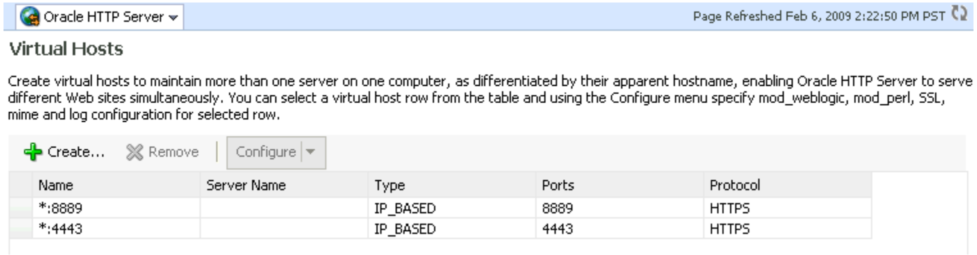
You can SSL-enable inbound traffic to Oracle HTTP Server virtual hosts using these steps:

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.

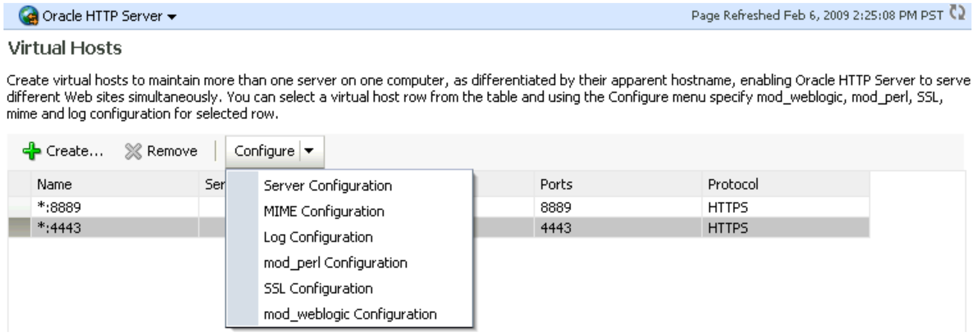


Virtual Hosts

Create virtual hosts to maintain more than one server on one computer, as differentiated by their apparent hostname, enabling Oracle HTTP Server to serve different Web sites simultaneously. You can select a virtual host row from the table and using the Configure menu specify mod_weblogic, mod_perl, SSL, mime and log configuration for selected row.

Name	Server Name	Type	Ports	Protocol
*:8889		IP_BASED	8889	HTTPS
*:4443		IP_BASED	4443	HTTPS

4. Select the virtual host you wish to update, and click **Configure**, then **SSL Configuration**.



Virtual Hosts

Create virtual hosts to maintain more than one server on one computer, as differentiated by their apparent hostname, enabling Oracle HTTP Server to serve different Web sites simultaneously. You can select a virtual host row from the table and using the Configure menu specify mod_weblogic, mod_perl, SSL, mime and log configuration for selected row.

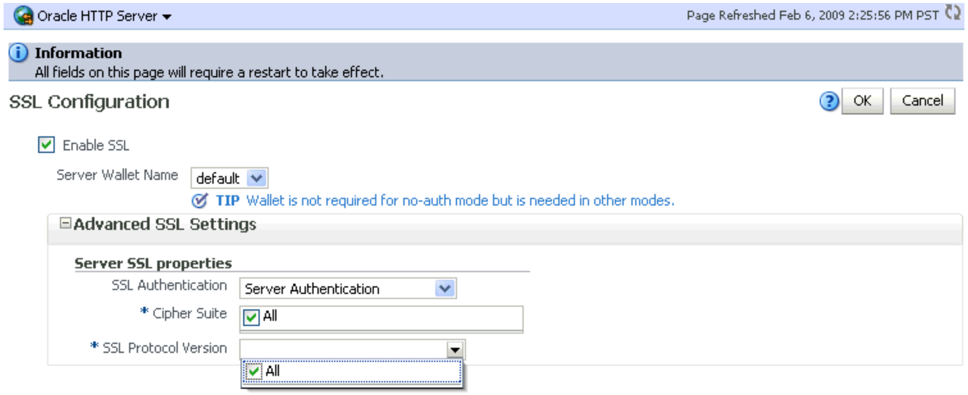
Name	Ser	Ports	Protocol
*:8889		8889	HTTPS
*:4443		4443	HTTPS

   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.



Oracle HTTP Server ▾ Page Refreshed Feb 6, 2009 2:25:56 PM PST

Information
All fields on this page will require a restart to take effect.

SSL Configuration

☑ Enable SSL

Server Wallet Name: default

TIP Wallet is not required for no-auth mode but is needed in other modes.

⊟Advanced SSL Settings

Server SSL properties

SSL Authentication: Server Authentication

* Cipher Suite: ☑ All

* SSL Protocol Version: ☑ All

- 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: 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 6.9.28, "Properties Files for SSL".

6. Click **OK** to apply the changes.
7. Restart the Oracle HTTP Server instance by navigating to **Oracle HTTP Server**, then **Control**, then **Restart.**
8. Open a browser session and connect to the port number that was SSL-enabled.

6.4.3.2 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.

2. Configure the virtual host with SSL properties:

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

Note:

- configureSSL uses defaults for all SSL attributes; see Table 6–5 for details.
- We could also specify a properties file as a parameter to configureSSL; see Table 6–4 for details.

6.4.3.3 Enable SSL for Outbound Requests from Oracle HTTP Server

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

The steps are as follows:

1. Generate a custom keystore for Oracle WebLogic Server (see Section 6.5.1, "Configuring SSL for Oracle WebLogic Server") 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.
3. Edit the Oracle HTTP Server configuration file *INSTANCE_HOME*/config/OHS/ohs1/ssl.conf and add the following line to the SSL configuration under mod_weblogic:

   ```
   WlSSLWallet  "$(ORACLE_INSTANCE}/config/COMPONENT_TYPE/COMPONENT_NAME/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 "$(ORACLE_INSTANCE)/config/OHS/ohs1/keystores/default"
   </IfModule>
   ```

   Save the file and exit. Restart Oracle HTTP Server to activate the changes.
4. 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.

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**.

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
- Configuring SSL for Oracle SOA Suite
- Configuring SSL for Oracle WebCenter
- Configuring SSL for Oracle Identity and Access Management
- SSL-Enable Oracle Reports, Forms, Discoverer, and Portal
- 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 *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

6.5.1.2 Outbound SSL from Oracle WebLogic Server

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

6.5.1.2.1 Outbound SSL from Oracle Platform Security Services to LDAP

This section explains how to configure SSL for policy stores and credential stores connections to an LDAP directory. Anonymous and one-way SSL is supported.

Anonymous SSL

Start the LDAP Server in anonymous authentication mode.

For Oracle Internet Directory, see Section 6.6.1.1, "Enable Inbound SSL on an Oracle Internet Directory Listener Using Fusion Middleware Control".

If using another directory, consult your LDAP server documentation for information on this task.

One-Way SSL

Prerequisite: LDAP Server in SSL Server Authentication Mode.

For details on this procedure, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

6.5.1.2.2 Outbound SSL from LDAP Authenticator to LDAP

When you configure an LDAP authenticator in Oracle WebLogic Server, you can specify that connections to the LDAP store should use SSL.

Take these steps to configure the authenticator:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the left pane, select **Security Realms** and click the name of the realm you are configuring.
3. Select **Providers**, then **Authentication** and click **New**.
4. In the **Name** field, enter a name for the authentication provider.
5. From the **Type** drop-down list, select the type of the Authentication provider and click **OK**.

   For example, if using Oracle Internet Directory, choose `OracleInternetDirectoryAuthenticator`.
6. Select **Providers**, then **Authentication** and click the name of the new authentication provider to complete its configuration.
7. On the Configuration page for the authentication provider, set the desired values on the **Common** and **Provider-Specific** tabs.
   a. Common Tab

      Set the Control Flag to `SUFFICIENT` for all authenticators, including the `DefaultAuthenticator`

   b. Provider-Specific Tab

      host: *host-name*

      port: *port-number*

      principal: cn=`orcladmin`

      credential/confirm: *password*

      user base dn: `cn=Users,dc=us,dc=oracle,dc=com`

      group base dn: `cn=Groups,dc=us,dc=oracle,dc=com`
8. Save your changes and restart the server.

6.5.1.2.3 Outbound SSL to Database

Configuring SSL between Oracle WebLogic Server and the database requires two sets of steps:

- Configuring SSL Listener for the Database
- Configuring SSL for the Data Source on Oracle WebLogic Server

Configure an SSL Listener on Oracle Database

To configure the database with an SSL listener, you must specify the server's distinguished name (DN) and TCPS as the protocol in the client network configuration files to enable server DN matching and TCP/IP with SSL connections. Server DN matching prevents the database server from faking its identity to the client during connections by matching the server's global database name against the DN from the server certificate.

You must manually edit the client network configuration files, `tnsnames.ora` and `listener.ora`, to specify the server's DN and the TCP/IP with SSL protocol.

For details, see Section 6.6.3.1, "SSL-Enable Oracle Database."

> **See Also:** Configuring Secure Sockets Layer Authentication in the *Oracle Database Advanced Security Administrator's Guide* for more information about configuring SSL for the database listener

SSL-Enable the Data Source On Oracle WebLogic Server

See Section 6.6.3.2, "SSL-Enable a Data Source."

6.5.2 Configuring SSL for Oracle SOA Suite

SSL configuration for Oracle SOA Suite varies with the type of connection being secured.

SSL in Oracle WebLogic Server

SSL features in Oracle WebLogic Server include:

- How to set up SSL at the core server
- How to enable SSL for a Web service

For these and related topics, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

SSL for SOA Composites

The following tasks are also needed to secure Oracle SOA Suite applications:

- SSL-protecting SOA composites
- Accessing SSL-protected Web services from within SOA composites

For these and related topics, see the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite*.

6.5.3 Configuring SSL for Oracle WebCenter

For information and details about how to implement SSL connections for Oracle WebCenter, see the following topics in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*:

- Securing the WebCenter Spaces Connection to Oracle Content Server with SSL
- Securing the Browser Connection to WebCenter Spaces with SSL

6.5.4 Configuring SSL for Oracle Identity and Access Management

You can configure SSL for Oracle Identity and Access Management components residing on the middle tier:

- Configuring SSL for Oracle Directory Integration Platform
- Configuring SSL for Oracle Identity Federation
- Configuring SSL for Oracle Directory Services Manager

6.5.4.1 Configuring SSL for Oracle Directory Integration Platform

You can configure Oracle Directory Integration Platform to use SSL for communications with connected directories. The *Oracle Fusion Middleware Integration Guide for Oracle Identity Management* provides details about the following SSL tasks for Oracle Directory Integration Platform:

- Configuring Oracle Directory Integration Platform for SSL Mode 2 Server-Only Authentication
- Managing the SSL Certificates of Oracle Internet Directory and Connected Directories
- Bootstrapping in SSL Mode
- Configuring the Third-Party Directory Connector for Synchronization in SSL Mode
- Configuring and Testing Oracle Internet Directory with SSL Server-Side Authentication
- Testing SSL Communication Between Oracle Internet Directory and Microsoft Active Directory

6.5.4.2 Configuring SSL for Oracle Identity Federation

See "Configuring SSL for Oracle Identity Federation" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Federation* for details.

> **Note:** Use Sun Microsystems' `keytool` utility to manage keystores and certificates required for SSL configuration in Oracle Identity Federation.

6.5.4.3 Configuring SSL for Oracle Directory Services Manager

You can configure Oracle Directory Services Manager to use SSL for communications with connected directories. The *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory* provides details about the following SSL tasks for Oracle Directory Services Manager:

- Logging into the Directory Server from Oracle Directory Services Manager Using SSL
- Managing Oracle Directory Services Manager's Key Store
- Storing Oracle Directory Services Manager's Certificate in Oracle Virtual Directory

6.5.5 SSL-Enable Oracle Reports, Forms, Discoverer, and Portal

This section contains these topics:

- SSL for Oracle Reports
- SSL for Oracle Forms
- SSL for Oracle Discoverer
- SSL for Oracle Portal

6.5.5.1 SSL for Oracle Reports

To SSL-enable Oracle Reports, you need to enable SSL on the components front-ending Oracle WebLogic Server.

For example, if you have an Oracle HTTP Server and an Oracle Web Cache front-ending the Oracle WebLogic Server that hosts Oracle Reports, you need to configure the following:

- Inbound SSL for Oracle Web Cache

  See Section 6.4.2.1, "Enable Inbound SSL for Oracle Web Cache Using Fusion Middleware Control."

- Inbound SSL for Oracle HTTP Server

  See Section 6.4.3.1, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control."

- Inbound SSL for Oracle WebLogic Server

  See Section 6.5.1.1, "Inbound SSL to Oracle WebLogic Server."

- SSL between Oracle Web Cache and Oracle HTTP Server

  See Section 6.4.2.3, "Enable Outbound SSL for Oracle Web Cache Using Fusion Middleware Control."

- SSL between Oracle HTTP Server and Oracle WebLogic Server

  See Section 6.4.3.3, "Enable SSL for Outbound Requests from Oracle HTTP Server."

> **Note:** These steps are necessary only if you wish to set up end-to-end SSL. In most cases, it is sufficient to enable SSL only on the first component getting the request, since the other components are usually within the intranet.
>
> For example, if the request is sent to Oracle Web Cache, you may only need to follow the first step. If the request is sent to Oracle HTTP Server, you may only need to follow the second step. Select the steps as dictated by your topology.

Additionally, Oracle Reports in Fusion Middleware Control accesses the reports servlet for data. If that communication needs to take place over SSL, you must complete the manual procedure described in *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*.

6.5.5.2 SSL for Oracle Forms

To SSL-enable Oracle Forms, you need to enable SSL on the components front-ending Oracle WebLogic Server.

For example, if you have an Oracle HTTP Server and an Oracle Web Cache front-ending the Oracle WebLogic Server that hosts Oracle Forms, you need to configure the following:

- Inbound SSL for Oracle Web Cache

  See Section 6.4.2.1, "Enable Inbound SSL for Oracle Web Cache Using Fusion Middleware Control."

- Inbound SSL for Oracle HTTP Server

  See Section 6.4.3.1, "Enable SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control."

- Inbound SSL for Oracle WebLogic Server

See Section 6.5.1.1, "Inbound SSL to Oracle WebLogic Server."

- SSL between Oracle Web Cache and Oracle HTTP Server

  See Section 6.4.2.3, "Enable Outbound SSL for Oracle Web Cache Using Fusion Middleware Control."

- SSL between Oracle HTTP Server and Oracle WebLogic Server

  See Section 6.4.3.3, "Enable SSL for Outbound Requests from Oracle HTTP Server."

> **Note:** These steps are necessary only if you wish to set up end-to-end SSL. In most cases, it is sufficient to enable SSL only on the first component getting the request, since the other components are usually within the intranet.
>
> For example, if the request is sent to Oracle Web Cache, you may only need to follow the first step. If the request is sent to Oracle HTTP Server, you may only need to follow the second step. Select the steps as dictated by your topology.

6.5.5.3 SSL for Oracle Discoverer

Running Oracle Discoverer over `https` requires installing a security certificate on the Discoverer Plus client machine, importing certificate details into the Java Plug-in certificate store, and related tasks.

The *Oracle Fusion Middleware Configuration Guide for Oracle Business Intelligence Discoverer* provides details about configuring SSL for Oracle Discoverer in these sections:

- About Running Discoverer over HTTPS
- About Discoverer and the Oracle Fusion Middleware Security Model

6.5.5.4 SSL for Oracle Portal

Oracle Portal uses a number of different components (such as the Parallel Page Engine, Oracle HTTP Server, and Oracle Web Cache) each of which may act as a client or server in HTTP communication. As a result, each component involving Oracle Portal in the middle tier is individually configured for https.

For details, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal*.

6.5.6 Client-Side SSL for Applications

For information on how to write SSL-enabled applications, see "Using SSL Authentication in Java Clients" in *Oracle Fusion Middleware Programming Security for Oracle WebLogic Server*.

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 these topics:

- Enabling SSL on Oracle Internet Directory Listeners
- Enabling SSL on Oracle Virtual Directory Listeners
- Configuring SSL for the Database

6.6.1 Enabling SSL on Oracle Internet Directory Listeners

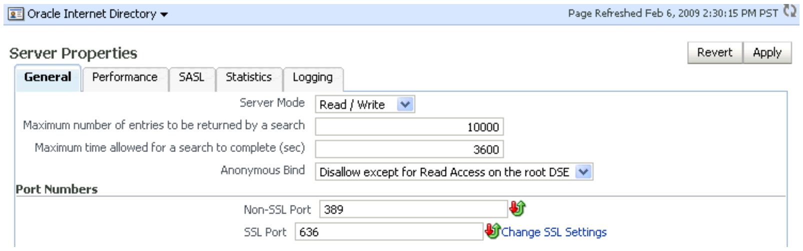
Out of the box, Oracle Internet Directory nodes are SSL-enabled in no-auth mode.

This section explains how to SSL-enable Oracle Internet Directory listeners using Fusion Middleware Control and the WLST command-line tool.

6.6.1.1 Enable Inbound SSL on an Oracle Internet Directory Listener Using Fusion Middleware Control

In this example, the following steps enable SSL in no-auth mode for an instance of Oracle Internet Directory using Fusion Middleware Control:

1. Select the Oracle Internet Directory instance in the navigation pane on the left.
2. Navigate to Oracle Internet Directory, then **Administration**, then **Server Properties**.



Oracle Internet Directory ▾
Page Refreshed Feb 6, 2009 2:30:15 PM PST
Revert
Apply
General
Performance
SASL
Statistics
Logging

Server Properties

Server Mode: Read / Write

Maximum number of entries to be returned by a search: 10000

Maximum time allowed for a search to complete (sec): 3600

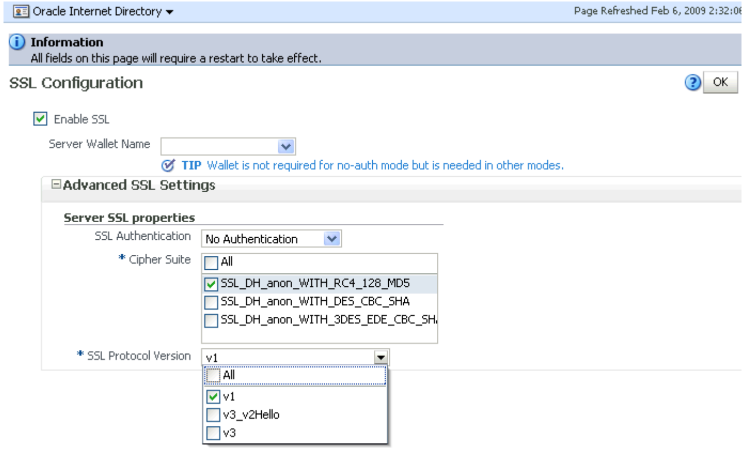
Anonymous Bind: Disallow except for Read Access on the root DSE

Port Numbers

Non-SSL Port: 389

SSL Port: 636

3. Click **Change SSL Settings**.
4. On the SSL Settings dialog:



Oracle Internet Directory ▾
Page Refreshed Feb 6, 2009 2:32:0
OK

Information
All fields on this page will require a restart to take effect.

SSL Configuration

☑ Enable SSL

Server Wallet Name:

TIP Wallet is not required for no-auth mode but is needed in other modes.

Advanced SSL Settings

Server SSL properties

SSL Authentication: No Authentication

* Cipher Suite:
☐ All
☑ SSL_DH_anon_WITH_RC4_128_MD5
☐ SSL_DH_anon_WITH_DES_CBC_SHA
☐ SSL_DH_anon_WITH_3DES_EDE_CBC_SH

* SSL Protocol Version: v1
☐ All
☑ v1
☐ v3_v2Hello
☐ v3

   - Select **Enable SSL**.
   - Set SSL Authentication to **No Authentication**.
   - Set Cipher Suite to **All**.
   - Set SSL protocol version to **v3**.
   - Click **OK**.

5. Restart the Oracle Internet Directory instance by navigating to **Oracle Internet Directory**, then **Control**, then **Restart.**
6. To verify that the instance is correctly SSL-enabled, execute an `ldapbind` command of the form:

```
ldapbind -D cn=orcladmin
   -U 1
   -h host
   -p SSL_port
```

Notes: `-U 1` represents the no-auth mode.

SSL Enabling in Other Authentication Modes

The steps for SSL-enabling in other authentication modes are the same, except that in the SSL Settings dialog, you would set the appropriate authentication type.

Note: Other authentication types need an Oracle wallet.

6.6.1.2 Enabling Inbound SSL on an Oracle Internet Directory Listener Using WLST

Configure the listener with SSL properties in `no-auth` mode as follows:

Note: The Oracle Internet Directory port name is always "`sslport1`".

```
configureSSL('inst1',
   'oid1',
   'oid',
   'sslport1')
```

Note:

- `configureSSL` can use defaults for all SSL attributes; see Table 6–5 for details.
- We could also specify a properties file as a parameter to `configureSSL`; see Table 6–4 for details.

SSL Enabling in Other Authentication Modes

You can do this by running the `configureSSL` command with a properties file as parameter and specifying an appropriate authentication type parameter value. For details, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

6.6.1.3 Enabling Outbound SSL from Oracle Internet Directory to Oracle Database

Two sets of procedures are needed to configure SSL connections from Oracle Internet Directory to Oracle Database:

- Configure SSL for the Database
- Configure Outbound Oracle Internet Directory

Configure SSL for the Database

The steps to configure Oracle Database for SSL are described in Section 6.6.3.1, "SSL-Enable Oracle Database."

Configure Outbound Oracle Internet Directory

Take these steps to configure SSL for outbound traffic from Oracle Internet Directory to Oracle Database:

1. Stop the Oracle Internet Directory server instances whose outbound traffic to the database is to be configured with SSL using this `opmnctl` syntax:

   ```
   $ORACLE_INSTANCE/bin/opmnctl stopproc ias-component=componentName
   ```

   For example:

   ```
   $ORACLE_INSTANCE/bin/opmnctl stopproc ias-component=oid1
   ```

2. Configure Security Socket Layer authentication on the database to which the Oracle Internet Directory server instance is connecting.

   For details, see *Oracle Database Advanced Security Administrator's Guide*.

3. Restart the database/listener as required.

4. Start Oracle Internet Directory server instances using this `opmnctl` syntax:

   ```
   $ORACLE_INSTANCE/bin/opmnctl startproc ias-component=componentName
   ```

   For example:

   ```
   $ORACLE_INSTANCE/bin/opmnctl startproc ias-component=oid1
   ```

   > **Note:** Only the no-authentication mode is supported.

6.6.2 Enabling SSL on Oracle Virtual Directory Listeners

This section explains how to enable SSL for an instance of Oracle Virtual Directory.

The *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory* provides additional information on these topics:

- Configuring SSL for Listeners Using Fusion Middleware Control
- Configuring SSL for Listeners Using WLST
- Configuring a Mutual Authentication SSL Connection Between Oracle Virtual Directory and Oracle Internet Directory

6.6.2.1 Enable SSL for Oracle Virtual Directory Using Fusion Middleware Control

The steps to enable SSL are as follows (the example illustrates the `server-auth` mode):

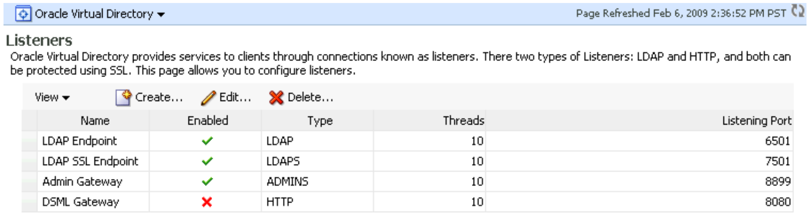
1. Select the Oracle Virtual Directory instance in the navigation pane on the left.

2. Select a keystore to use for the operation by navigating to **Oracle Virtual Directory**, then **Security**, then **Keystores**

   Choose from the list of keystores that appears. If you need to generate a new keystore, see Section 7.3.3.1, "Creating a Keystore Using Fusion Middleware Control" for details.

3. To SSL-enable the listener, navigate to **Oracle Virtual Directory**, then **Administration**, then **Listeners**.

4. Select the LDAP SSL Endpoint listener, and click **Edit**.

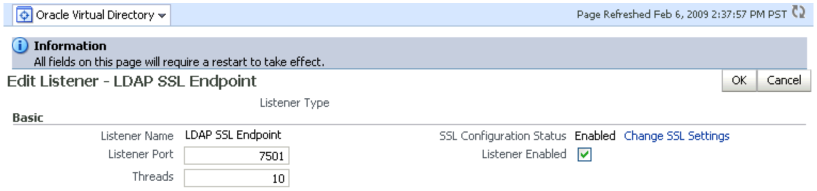


Listeners

Oracle Virtual Directory provides services to clients through connections known as listeners. There two types of Listeners: LDAP and HTTP, and both can be protected using SSL. This page allows you to configure listeners.

Name	Enabled	Type	Threads	Listening Port
LDAP Endpoint	✓	LDAP	10	6501
LDAP SSL Endpoint	✓	LDAPS	10	7501
Admin Gateway	✓	ADMINS	10	8899
DSML Gateway	✗	HTTP	10	8080

The Edit Listener page appears:



Information
All fields on this page will require a restart to take effect.

Edit Listener - LDAP SSL Endpoint

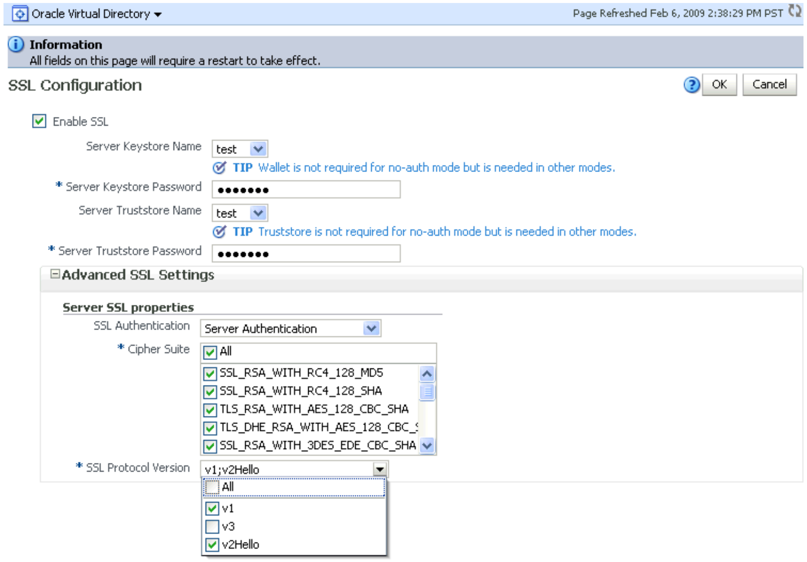
Listener Type

Basic

Listener Name LDAP SSL Endpoint
Listener Port 7501
Threads 10
SSL Configuration Status Enabled Change SSL Settings
Listener Enabled ☑

5. Click **Change SSL Settings**.

6. On the SSL Settings dialog:



Information
All fields on this page will require a restart to take effect.

SSL Configuration

☑ Enable SSL

Server Keystore Name test
TIP Wallet is not required for no-auth mode but is needed in other modes.
* Server Keystore Password •••••••
Server Truststore Name test
TIP Truststore is not required for no-auth mode but is needed in other modes.
* Server Truststore Password •••••••

⊟ Advanced SSL Settings

Server SSL properties

SSL Authentication Server Authentication
* Cipher Suite ☑ All
☑ SSL_RSA_WITH_RC4_128_MD5
☑ SSL_RSA_WITH_RC4_128_SHA
☑ TLS_RSA_WITH_AES_128_CBC_SHA
☑ TLS_DHE_RSA_WITH_AES_128_CBC_S
☑ SSL_RSA_WITH_3DES_EDE_CBC_SHA
* SSL Protocol Version v1;v2Hello
□ All
☑ v1
□ v3
☑ v2Hello

- Select **Enable SSL**.
- For Server Keystore Name, select the keystore you created in step 3, for example, `OVDtestJks`.
- For Server Keystore Password, type the keystore password you specified in step 3.
- For Server Truststore Name, select the keystore you created in step 3, for example, `OVDtestJks`.
- For Server Truststore Password, type the keystore password you specified in step 3.
- Expand Advanced SSL Settings.
- For SSL authentication, select **Server Authentication**. This is the default setting.
- For Cipher Suite, select the applicable cipher suite, in this example `All`.

- Click **OK**.

7. Stop and start the Oracle Virtual Directory instance by navigating to **Oracle Virtual Directory**, then **Control**, then **Stop** and **Start.**

8. To verify that the instance is correctly SSL-enabled, execute an `ldapbind` command of the form:

   ```
   ldapbind -D cn=orcladmin
      -U 2
      -h host
      -p SSL_port
   -W "file:// DIRECTORY_SSL_WALLET"
   ```

   > **Note:**
   >
   > - `-U 2` represents the server-auth mode.
   > - `DIRECTORY_SSL_WALLET` is the path to a wallet file, not including the wallet file name.
   > - This wallet must contain the trusted certificate of the CA that issued the server certificate.

SSL Enabling in Other Authentication Modes

The steps for SSL-enabling in other authentication modes are similar, except that in the SSL Settings dialog, you would set the appropriate authentication type.

> **Note:** If configuring SSL for an LDAP listener, SSL communication is verified using `ldapbind`. If it is an http listener, it is verified using a browser.

6.6.2.2 Enabling SSL on an Oracle Virtual Directory Listener Using WLST

Take these steps to configure the listener in server-auth mode:

1. Determine the listeners for this Oracle Virtual Directory instance by running the following command:

   ```
   listListeners('inst1','ovd1' )
   ```

   This command lists all the listeners for this instance; select the one that needs to be configured for SSL. For this example, select **LDAP SSL Endpoint**.

2. Obtain the name of the SSL MBean for the Oracle Virtual Directory listener:

   ```
   getSSLMBeanName('inst1',
      'ovd1',
      'ovd',
      'LDAP SSL Endpoint')
   ```

   This command will return the SSL MBean name.

3. Set the passwords for the keystore and truststore in the MBean with the following commands:

   ```
   cd ('SSL_MBean_Name')
   set('KeyStorePassword',java.lang.String('password').toCharArray())
   set('TrustStorePassword',java.lang.String('password').toCharArray())
   ```

4. Configure the listener with SSL properties:

```
configureSSL('inst1',
   'ovd1',
   'ovd',
   'LDAP SSL Endpoint')
```

> **Note:** Steps 2 and 3 are required only for server-auth and mutual-auth modes.

Enabling SSL in Other Authentication Modes

You can do this by running the `configureSSL` command with a properties file as parameter and specifying appropriate authentication type parameter value. For details, see "Creating and Managing Oracle Virtual Directory Listeners" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*.

6.6.3 Configuring SSL for the Database

This section contains these topics:

- SSL-Enable Oracle Database
- SSL-Enable a Data Source

6.6.3.1 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)
    )
    (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*.

6.6.3.2 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 data source, or a new data source. See Creating a JDBC Data Source in *Oracle Fusion Middleware Configuring and Managing JDBC 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:

   ```
   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")))
   ```

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

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, Oracle Web Cache, and Oracle Internet Directory, which are the system components 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 Oracle Wallet Manager (OWM) or 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, that is, creating a certificate request, getting the request approved by a Certificate Authority (CA), and installing 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 H, "Oracle Wallet Manager and 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 and Oracle Web Cache.
> - CRL validation is managed 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 H.2.5, "Managing Certificate Revocation Lists (CRLs) with orapki Utility."

CRL Renaming to Hashed Form

If specifying a fleshiest directory, 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, the CRL is copied 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.txt -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.

6.8 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 hostname 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

Starting with 11*g* Release 1 (11.1.1), WLST commands have been added to manage Oracle wallets and JKS keystores and to configure SSL for Oracle Fusion Middleware components.

Use the commands listed in Table 6–1,Table 6–2, and Table 6–3 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')
```

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

Table 6–3 WLST Commands for Java Keystore (JKS) Management

Use this command...	To...	Use with WLST...
changeKeyStorePassword	Change the password to a JKS keystore.	Online
createKeyStore	Create a JKS keystore.	Online
deleteKeyStore	Delete a JKS keystore.	Online
exportKeyStore	Export a JKS keystore to a file.	Online
exportKeyStoreObject	Export an object (for example, a certificate) from a JKS keystore to a file.	Online
generateKey	Generate a keypair in a JKS keystore.	Online
getKeyStoreObject	Display a certificate or other object present in a JKS keystore.	Online
importKeyStore	Import a JKS keystore from a file.	Online
importKeyStoreObject	Import a certificate or other object from a file to a JKS keystore.	Online
listKeyStoreObjects	List all objects (for example, certificates) present in a JKS keystore.	Online
listKeyStores	List all JKS keystores configured for a component instance.	Online
removeKeyStoreObject	Remove a certificate or other object from a component instance's JKS keystore.	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, Oracle WebCache or Oracle Internet Directory). 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
instName	Specifies the name of the application server instance.
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid values are 'ohs', 'oid', and 'webcache'.
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.

6.9.1.3 Example

The following command generates a certificate signing request with DN `cn=www.acme.com` and key size `1024` in `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> addCertificateRequest('inst1', 'oid1',
'oid','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, Oracle WebCache or Oracle Internet Directory). 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 values are 'ohs', 'oid', and 'webcache'.
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 Internet Directory instance `oid1`, in application server instance inst1:

```
wls:/mydomain/serverConfig> addSelfSignedCertificate('inst1', 'oid1',
'oid','wallet1', 'password', 'cn=www.acme.com', '1024')
```

6.9.3 changeKeyStorePassword

Online command that changes the keystore password.

6.9.3.1 Description

This command changes the password of a Java Keystore (JKS) file for an Oracle Virtual Directory instance.

6.9.3.2 Syntax

```
changeKeyStorePassword('instName', 'compName', 'compType', 'keystoreName',
'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 'ovd'.
keystoreName	Specifies the filename of the keystore.
currPassword	Specifies the current keystore password.
newPassword	Specifies the new keystore password.

6.9.3.3 Example

The following command changes the password of file `keys.jks` for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig> changeKeyStorePassword('inst1', 'ovd1',
'ovd','keys.jks', 'currpassword', 'newpassword')
```

6.9.4 changeWalletPassword

Online command that changes the password of an Oracle wallet.

6.9.4.1 Description

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

6.9.4.2 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 values are 'oid', 'ohs', and 'webcache'.
walletName	Specifies the filename of the wallet.
currPassword	Specifies the current wallet password.
newPassword	Specifies the new wallet password.

6.9.4.3 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')
```

6.9.5 configureSSL

Online command that sets SSL attributes.

6.9.5.1 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, default attribute values are used.

For details about the format of properties files, see Section 6.9.28, "Properties Files for SSL."

6.9.5.2 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 values are 'oid', 'ovd', ohs', and 'webcache'.
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.

6.9.5.3 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')
```

6.9.6 createKeyStore

Online command that creates a JKS keystore.

6.9.6.1 Description

This command creates a Java keystore (JKS) for the specified Oracle Virtual Directory instance. For keystore file location and other information, see Section 7.3.6.1, "Location of Keystores."

6.9.6.2 Syntax

```
createKeyStore('instName', 'compName', 'compType', 'keystoreName', 'password')
```

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 'ovd'.
keystoreName	Specifies the filename of the keystore file to be created.
password	Specifies the keystore password.

6.9.6.3 Example

The following command creates JKS file `keys.jks` with password *`password`* for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig> createKeyStore('inst1', 'ovd1', 'ovd','keys.jks',
'password')
```

6.9.7 createWallet

Online command that creates an Oracle wallet.

6.9.7.1 Description

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

6.9.7.2 Syntax

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

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 values are 'oid', 'ohs', and 'webcache'.
walletName	Specifies the name of the wallet file to be created.
password	Specifies the wallet password.

6.9.7.3 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', '')
```

6.9.8 deleteKeyStore

Online command that deletes a keystore.

6.9.8.1 Description

This command deletes a keystore for a specified Oracle Virtual Directory instance.

6.9.8.2 Syntax

```
deleteKeyStore('instName', 'compName', 'compType', 'keystoreName')
```

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 'ovd'.
keystoreName	Specifies the name of the keystore file to delete.

6.9.8.3 Example

The following command deletes JKS file `keys.jks` for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig> deleteKeyStore('inst1', 'ovd1', 'ovd','keys.jks')
```

6.9.9 deleteWallet

Online command that deletes an Oracle wallet.

6.9.9.1 Description

This command deletes an Oracle wallet for the specified component instance (Oracle HTTP Server, Oracle WebCache or Oracle Internet Directory).

6.9.9.2 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 values are 'oid', 'ohs', and 'webcache'.
walletName	Specifies the name of the wallet file to be deleted.

6.9.9.3 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')
```

6.9.10 exportKeyStore

Online command that exports the keystore to a file.

6.9.10.1 Description

This command exports a keystore, configured for the specified Oracle Virtual Directory instance, to a file under the given directory. The exported filename is the same as the keystore name.

6.9.10.2 Syntax

```
exportKeyStore('instName', 'compName', 'compType', 'keystoreName',
               '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 'ovd'.
keystoreName	Specifies the name of the keystore file.
password	Specifies the password of the keystore.
path	Specifies the absolute path of the directory under which the keystore is exported.

6.9.10.3 Example

The following command exports the keystore `keys.jks` for Oracle Virtual Directory instance `ovd1` to file `keys.jks` under `/tmp`:

```
wls:/mydomain/serverConfig> exportKeyStore('inst1', 'ovd1', 'ovd', 'keys.jks',
'password', '/tmp')
```

6.9.11 exportKeyStoreObject

Online command that exports an object from a keystore to a file.

6.9.11.1 Description

This command exports a certificate signing request, certificate/certificate chain, or trusted certificate present in a Java keystore (JKS) to a file for the specified Oracle Virtual Directory instance. The certificate signing request is generated before exporting the object. The alias specifies the object to be exported.

6.9.11.2 Syntax

```
exportKeyStoreObject('instName', 'compName', 'compType', 'keystoreName',
'password', 'type', 'path', 'alias')
```

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 'ovd'.
keystoreName	Specifies the name of the keystore file.
password	Specifies the password of the keystore.
type	Specifies the type of the keystore object to be exported. Valid values are 'CertificateRequest', 'Certificate', 'TrustedCertificate' and 'TrustedChain'.
path	Specifies the absolute path of the directory under which the object is exported as a file named base64.txt.
alias	Specifies the alias of the keystore object to be exported.

6.9.11.3 Examples

The following command generates and exports a certificate signing request from the key-pair indicated by alias `mykey` in `keys.jks`, for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`. The certificate signing request is exported under the directory `/tmp`:

```
wls:/mydomain/serverConfig> exportKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'CertificateRequest', '/tmp','mykey')
```

The following command exports a certificate or certificate chain indicated by alias `mykey` in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`. The certificate or certificate chain is exported under the directory `/tmp`:

```
wls:/mydomain/serverConfig> exportKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'Certificate', '/tmp','mykey')
```

The following command exports a trusted certificate indicated by alias `mykey` in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`. The trusted certificate is exported under the directory `/tmp`:

```
wls:/mydomain/serverConfig> exportKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'TrustedCertificate', '/tmp','mykey')
```

6.9.12 exportWallet

Online command that exports an Oracle wallet.

6.9.12.1 Description

This command exports an Oracle wallet, configured for a specified component instance (Oracle HTTP Server, Oracle WebCache or Oracle Internet Directory), 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.

6.9.12.2 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 values are 'oid', 'ohs', and 'webcache'.
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.

6.9.12.3 Examples

The following command exports auto-login wallet `wallet1` for Oracle Internet Directory instance `oid1` to file `cwallet.sso` under `/tmp`:

```
wls:/mydomain/serverConfig> exportWallet('inst1', 'oid1', 'oid',
'wallet1','','/tmp')
```

The following command exports password-protected wallet `wallet2` for Oracle Internet Directory instance `oid1` to two files, `ewallet.p12` and `cwallet.sso`, under `/tmp`:

```
wls:/mydomain/serverConfig> exportWallet('inst1', 'oid1', 'oid', 'wallet2',
'password', '/tmp')
```

6.9.13 exportWalletObject

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

6.9.13.1 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 (Oracle HTTP Server, Oracle WebCache or Oracle Internet Directory). DN indicates the object to be exported.

6.9.13.2 Syntax

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

Argument	Definition
instName	Specifies the name of the application server instance.

Argument	Definition
compName	Specifies the name of the component instance.
compType	Specifies the type of component. Valid values are 'ohs', 'oid', and 'webcache'.
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.

6.9.13.3 Examples

The following command exports a certificate signing request with DN cn=www.acme.com in wallet1, for Oracle Internet Directory instance oid1, in application server instance inst1. The certificate signing request is exported under the directory /tmp:

```
wls:/mydomain/serverConfig> exportWalletObject('inst1', 'oid1', 'oid','wallet1', 
'password', 'CertificateRequest', '/tmp','cn=www.acme.com')
```

The following command exports a certificate with DN cn=www.acme.com in wallet1, for Oracle Internet Directory instance oid1, in application server instance inst1. The certificate or certificate chain is exported under the directory /tmp:

```
wls:/mydomain/serverConfig> exportWalletObject('inst1', 'oid1', 'oid','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 Internet Directory instance oid1, in application server instance inst1. The trusted certificate is exported under the directory /tmp:

```
wls:/mydomain/serverConfig> exportWalletObject('inst1', 'oid1', 'oid','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 Internet Directory instance oid1, in application server instance inst1. The certificate or certificate chain is exported under the directory /tmp:

```
wls:/mydomain/serverConfig> exportWalletObject('inst1', 'oid1', 'oid','wallet1', 
'password', 'TrustedChain', '/tmp','cn=www.acme.com')
```

6.9.14 generateKey

Online command that generates a key pair in a Java keystore.

6.9.14.1 Description

This command generates a key pair in a Java keystore (JKS) for Oracle Virtual Directory. It also wraps the key pair in a self-signed certificate. Only keys based on the RSA algorithm are generated.

6.9.14.2 Syntax

```
generateKey('instName', 'compName', 'compType', 'keystoreName', 'password', 'DN',
'keySize', 'alias', 'algorithm')
```

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 'ovd'.
keystoreName	Specifies the name of the keystore.
password	Specifies the password of the keystore.
DN	Specifies the Distinguished Name of the key pair entry.
keySize	Specifies the key size in bits.
alias	Specifies the alias of the key pair entry in the keystore.
algorithm	Specifies the key algorithm. Valid value is 'RSA'.

6.9.14.3 Examples

The following command generates a key pair with DN `cn=www.acme.com`, key size `1024`, algorithm `RSA` and alias `mykey` in `keys.jks`, for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig> generateKey('inst1', 'ovd1', 'ovd','keys.jks',
'password', 'cn=www.acme.com', '1024', 'mykey', 'RSA')
```

The following command is the same as above, except it does not explicitly specify the key algorithm:

```
wls:/mydomain/serverConfig> generateKey('inst1', 'ovd1', 'ovd','keys.jks',
'password', 'cn=www.acme.com', '1024', 'mykey')
```

6.9.15 getKeyStoreObject

Online command that shows details about a keystore object.

6.9.15.1 Description

This command displays a specific certificate or trusted certificate present in a Java keystore (JKS) for Oracle Virtual Directory. The keystore object is indicated by its index number, as given by the `listKeyStoreObjects` command. It shows the certificate details including DN, key size, algorithm, and other information.

6.9.15.2 Syntax

```
getKeyStoreObject('instName', 'compName', 'compType', 'keystoreName', '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 'ovd'.
keystoreName	Specifies the name of the keystore file.
password	Specifies the password of the keystore.

Argument	Definition
type	Specifies the type of the keystore object to be listed. Valid values are 'Certificate' and 'TrustedCertificate'.
index	Specifies the index number of the keystore object as returned by the `listKeyStoreObjects` command.

6.9.15.3 Examples

The following command shows a trusted certificate with index `1` present in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> getKeyStoreObject('inst1', 'ovd1', 'ovd','keys.jks',
'password', 'TrustedCertificate', '1')
```

The following command shows a certificate with index `1` present in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> getKeyStoreObject('inst1', 'ovd1', 'ovd','keys.jks',
'password', 'Certificate', '1')
```

6.9.16 getSSL

Online command that lists the configured SSL attributes.

6.9.16.1 Description

This command lists the configured SSL attributes for the specified component listener. For Oracle Internet Directory, the listener name is always `sslport1`.

6.9.16.2 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 values are 'ovd', 'oid', 'ohs', and 'webcache'.
listener	Specifies the name of the component listener.

6.9.16.3 Example

The following command shows the SSL attributes configured for Oracle Internet Directory instance `oid1`, in application server instance `inst1`, for listener `sslport1`:

```
wls:/mydomain/serverConfig> getSSL('inst1', 'oid1', 'oid', 'sslport1')
```

6.9.17 getWalletObject

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

6.9.17.1 Description

This command displays a specific certificate signing request, certificate or trusted certificate present in an Oracle wallet for the specified component instance (Oracle HTTP Server, Oracle WebCache or Oracle Internet Directory). 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.

6.9.17.2 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 values are 'ohs', 'oid', and 'webcache'.
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'.
index	Specifies the index number of the wallet object as returned by the `listWalletObjects` command.

6.9.17.3 Examples

The following command shows certificate signing request details for the object with index `0` present in `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> getKeyStoreObject('inst1', 'oid1',
'oid','wallet1','password', 'CertificateRequest', '0')
```

The following command shows certificate details for the object with index `0` present in `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> getKeyStoreObject('inst1', 'oid1',
'oid','wallet1','password', 'Certificate', '0')
```

The following command shows trusted certificate details for the object with index `0`, present in `wallet1`, for Oracle Internet Directory instance `oid1`, in application serverinstance `inst1`:

```
wls:/mydomain/serverConfig> getKeyStoreObject('inst1', 'oid1',
'oid','wallet1','password', 'TrustedCertificate', '0')
```

6.9.18 importKeyStore

Online command that imports a keystore from a file.

6.9.18.1 Description

This command imports a Java keystore (JKS) from a file to the specified Oracle Virtual Directory instance for manageability. The component instance name must be unique.

6.9.18.2 Syntax

```
importKeyStore('instName', 'compName', 'compType', 'keystoreName',
'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 'ovd'.
keystoreName	Specifies the name of the keystore being imported. This name must be unique for this component instance.
password	Specifies the password of the keystore.
filePath	Specifies the absolute path of the keystore file to be imported.

6.9.18.3 Example

The following command imports the keystore `/tmp/keys.jks` as `file.jks` into Oracle Virtual Directory instance `ovd1`. Subsequently, the keystore is managed through the name `file.jks`:

```
wls:/mydomain/serverConfig> importKeyStore('inst1', 'ovd1', 'ovd', 'file.jks',
'password', '/tmp/keys.jks')
```

6.9.19 importKeyStoreObject

Online command that imports an object from a file to a keystore.

6.9.19.1 Description

This command imports a certificate, certificate chain, or trusted certificate into a Java keystore (JKS) for Oracle Virtual Directory, assigning it the specified alias which must be unique in the keystore. If a certificate or certificate chain is being imported, the alias must match that of the corresponding key-pair.

6.9.19.2 Syntax

```
importKeyStoreObject('instName', 'compName', 'compType', 'keystoreName',
'password', 'type', 'filePath', 'alias')
```

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 'ovd'.
keystoreName	Specifies the name of the keystore.
password	Specifies the password of the keystore.
type	Specifies the type of the keystore object to be imported. Valid values are 'Certificate' and 'TrustedCertificate'.
filePath	Specifies the absolute path of the file containing the keystore object.

Argument	Definition
alias	Specifies the alias to assign to the keystore object to be imported.

6.9.19.3 Examples

The following command imports a certificate or certificate chain from file `cert.txt` into `keys.jks`, using alias `mykey` for Oracle Virtual Directory instance `ovd1,` in application server instance `inst1`. The file `keys.jks` must already have an alias `mykey` for a key-pair whose public key matches that in the certificate being imported:

```
wls:/mydomain/serverConfig> > importKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'Certificate','/tmp/cert.txt', 'mykey')
```

The following command imports a trusted certificate from file `trust.txt` into `keys.jks` using alias `mykey1`, for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig> importKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'TrustedCertificate','/tmp/trust.txt', 'mykey1')
```

6.9.20 importWallet

Online command that imports an Oracle wallet from a file.

6.9.20.1 Description

This command imports an Oracle wallet from a file to the specified component instance (Oracle HTTP Server, Oracle WebCache, or Oracle Internet Directory) 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.

6.9.20.2 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 values are 'ohs', 'oid', and 'webcache'.
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.

6.9.20.3 Examples

The following command imports the auto-login wallet file `/tmp/cwallet.sso` as `wallet1` into Oracle Internet Directory instance `oid1`. 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', 'oid1', 'oid', 'wallet1', '',
'/tmp/cwallet.sso')
```

The following command imports password-protected wallet `/tmp/ewallet.p12` as `wallet2` into Oracle Internet Directory instance `oid1`. Subsequently, the wallet is managed with the name `wallet2`. The wallet password is passed as a parameter:

```
wls:/mydomain/serverConfig> importWallet('inst1', 'oid1', 'oid', 'wallet2',
'password', '/tmp/ewallet.p12')
```

6.9.21 importWalletObject

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

6.9.21.1 Description

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

6.9.21.2 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 values are 'ohs', 'oid', and 'webcache'.
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.

6.9.21.3 Examples

The following command imports a certificate chain in PKCS#7 format from file `chain.txt` into `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> importWalletObject('inst1', 'oid1', 'oid','wallet1',
'password', 'TrustedChain','/tmp/chain.txt')
```

The following command imports a certificate from file `cert.txt` into `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> > importWalletObject('inst1', 'oid1', 'oid','wallet1',
'password', 'Certificate','/tmp/cert.txt')
```

The following command imports a trusted certificate from file `trust.txt` into `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> importWalletObject('inst1', 'oid1', 'oid','wallet1',
'password', 'TrustedCertificate','/tmp/trust.txt')
```

6.9.22 listKeyStoreObjects

Online command that lists the contents of a keystore.

6.9.22.1 Description

This command lists all the certificates or trusted certificates present in a Java keystore (JKS) for Oracle Virtual Directory.

6.9.22.2 Syntax

```
listKeyStoreObjects('instName', 'compName', 'compType', 'keystoreName',
'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 'ovd'.
keystoreName	Specifies the name of the keystore file.
password	Specifies the password of the keystore.
type	Specifies the type of keystore object to be listed. Valid values are 'Certificate' and 'TrustedCertificate'.

6.9.22.3 Examples

The following command lists all trusted certificates present in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> listKeyStoreObjects('inst1', 'ovd1', 'ovd','keys.jks',
'password', 'TrustedCertificate')
```

The following command lists all certificates present in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> listKeyStoreObjects('inst1', 'ovd1', 'ovd','keys.jks',
'password', 'Certificate')
```

6.9.23 listKeyStores

Online command that lists all the keystores for a component.

6.9.23.1 Description

This command lists all the Java keystores (JKS) configured for the specified Oracle Virtual Directory instance.

6.9.23.2 Syntax

```
listKeyStores('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 'ovd'.

6.9.23.3 Example

The following command lists all keystores for Oracle Virtual Directory instance `ovd1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig> listKeyStores('inst1', 'ovd1', 'ovd')
```

6.9.24 listWalletObjects

Online command that lists all objects in an Oracle wallet.

6.9.24.1 Description

This command lists all certificate signing requests, certificates, or trusted certificates present in an Oracle wallet for the specified component instance (Oracle HTTP Server, Oracle WebCache or Oracle Internet Directory).

6.9.24.2 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 values are 'ohs','oid', and 'webcache'.
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'.

6.9.24.3 Examples

The following command lists all certificate signing requests in `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> > listWalletObjects('inst1', 'oid1',
'oid','wallet1','password', 'CertificateRequest')
```

The following command lists all certificates in `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> listWalletObjects('inst1', 'oid1',
'oid','wallet1','password', 'Certificate')
```

The following command lists all trusted certificates in `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> listWalletObjects('inst1', 'oid1',
'oid','wallet1','password', 'TrustedCertificate')
```

6.9.25 listWallets

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

6.9.25.1 Description

This command displays all the wallets configured for the specified component instance (Oracle HTTP Server, Oracle WebCache or Oracle Internet Directory), and identifies the auto-login wallets.

6.9.25.2 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 values are 'ohs','oid', and 'webcache'.

6.9.25.3 Example

The following command lists all wallets for Oracle Internet Directory instance `oid1` in application server instance `inst1`:

```
wls:/mydomain/serverConfig> > listWallets('inst1', 'oid1', 'oid')
```

6.9.26 removeKeyStoreObject

Online command that removes an object from a keystore.

6.9.26.1 Description

This command removes a certificate request, certificate, trusted certificate, or all trusted certificates from a Java keystore (JKS) for Oracle Virtual Directory. Use an alias to remove a specific object; no alias is needed if all trusted certificates are being removed.

6.9.26.2 Syntax

```
removeKeyStoreObject('instName', 'compName', 'compType', 'keystoreName',
'password', 'type', 'alias')
```

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 'ovd'.
keystoreName	Specifies the name of the keystore file.
password	Specifies the password of the keystore.
type	Specifies the type of the keystore object to be removed. Valid values are 'Certificate', 'TrustedCertificate' or 'TrustedAll'.
alias	Specifies the alias of the keystore object to be removed.

6.9.26.3 Examples

The following command removes a certificate or certificate chain denoted by alias `mykey` in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> removeKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'Certificate','mykey')
```

The following command removes a trusted certificate denoted by alias `mykey` in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> removeKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'TrustedCertificate','mykey')
```

The following command removes all trusted certificates in `keys.jks`, for Oracle Virtual Directory instance `ovd1`, in application server instance `inst1`. Since no alias is required, the value `None` is passed for that parameter:

```
wls:/mydomain/serverConfig> removeKeyStoreObject('inst1', 'ovd1',
'ovd','keys.jks', 'password', 'TrustedAll',None)
```

6.9.27 removeWalletObject

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

6.9.27.1 Description

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

6.9.27.2 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 values are 'ohs','oid', and 'webcache'.
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.

6.9.27.3 Examples

The following command removes all trusted certificates from `wallet1`, for Oracle Internet Directory instance `oid1`, 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', 'oid1', 'oid','wallet1',
'password', 'TrustedAll',None)
```

The following command removes a certificate signing request indicated by DN `cn=www.acme.com` from `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> removeWalletObject('inst1', 'oid1', 'oid','wallet1',
'password', 'CertificateRequest','cn=www.acme.com')
```

The following command removes a certificate indicated by DN `cn=www.acme.com` from `wallet1`, for Oracle Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> removeWalletObject('inst1', 'oid1', 'oid','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 Internet Directory instance `oid1`, in application server instance `inst1`:

```
wls:/mydomain/serverConfig> removeWalletObject('inst1', 'oid1', 'oid','wallet1',
'password', 'TrustedCertificate','cn=www.acme.com')
```

6.9.28 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`.

6.9.28.1 Structure of Properties Files

All the SSL properties files have a consistent structure.

Table 6–4 provides details about the key-value structure and usage of these files.

Table 6–4 Parameters in Properties File

Key	Mandatory?	Allowed Values for Oracle HTTP Server, Oracle Internet Directory, and Oracle Web Cache	Allowed Values for Oracle Virtual Directory	Usage
SSLEnabled	No	true false	true false	Either value
Ciphers	No	SSL_RSA_WITH_RC4_128_MD5 SSL_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_3DES_EDE_CBC_SHA SSL_RSA_WITH_DES_CBC_SHA SSL_DH_anon_WITH_RC4_128_MD5 SSL_DH_anon_WITH_DES_CBC_SHA SSL_DH_anon_WITH_3DES_EDE_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA	One of more of the ciphers allowed by the JSSE provider. For the complete list of ciphers allowed by JDK 1.5, see Appendix A of the following guide -`http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html`	One or more comma separated values
SSLVersions	No	nzos_Version_3_0 nzos_Version_3_0_With_2_0_Hello nzos_Version_1_0	TLSv1 SSLv2Hello (cannot be specified alone, must specify at least one other version) SSLv3	One or more comma separated values

Table 6–4 (Cont.) Parameters in Properties File

Key	Mandatory?	Allowed Values for Oracle HTTP Server, Oracle Internet Directory, and Oracle Web Cache	Allowed Values for Oracle Virtual Directory	Usage
CertValidation	No	none crl	N/A	Either value
CertValidation Path	No	file://`crl_file_path` dir://`crl_dir_path`	N/A	Path of the CRL file, or directory containing CRL files
KeyStore	No	Valid wallet name	Valid keystore name	
TrustStore	No	N/A	Valid truststore name	
AuthenticationType	No	None Server Optional Mutual	None Server Optional Mutual	Any one value

Table 6–5 shows the default values:

Table 6–5 Default Values of Parameters

Key	Default Value for Oracle HTTP Server	Default Value for Oracle Web Cache	Default Value for Oracle Internet Directory	Default Value for Oracle Virtual Directory
SSLEnabled	true	true	true	true
Ciphers	null	null	null	null
SSLVersions	null	null	null	null
CertValidation	none	none	-	-
CertValidation Path	null	null	-	-
KeyStore	default	default	null	keys.jks
TrustStore	-	-	-	keys.jks
Authentication Type	Server	Server	none	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.

6.9.28.2 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:

```
SSLEnabled=true
```

Here the value `true` is explicitly specified.

```
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_3_0
CertValidation=crl
CertValidationPath=file:///tmp/file.crl
KeyStore=ohs1
```

This properties file has:

- Default values for ciphers
- Keystore
- SSL version v3
- CRL validation turned on
- Mutual Authentication mode

Example 4: Properties File with Ciphers for Oracle Virtual Directory

```
AuthenticationType=Server
Ciphers=SSL_RSA_WITH_RC4_128_MD5
```

```
SSLVersion=SSLv3,SSLv2Hello
KeyStore=ovdidentity.jks
TrustStore=ovdtrust.jks
SSLEnabled=true
```

This properties file contains:

- Specific cipher value
- SSL Version
- Server authentication mode

7

Managing Keystores, Wallets, and Certificates

This chapter explains how to use Oracle Fusion Middleware security features to administer keystores, wallets, and certificates. It contains these sections:

- Key and Certificate Storage in Oracle Fusion Middleware
- Command-Line Interface for Keystores and Wallets
- JKS Keystore Management
- Wallet Management

7.1 Key and Certificate Storage in Oracle Fusion Middleware

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 two types of keystores for keys and certificates:

- JKS Keystore and Truststore
- Oracle Wallet

7.1.1.1 JKS Keystore and Truststore

A JKS keystore is the default JDK implementation of Java keystores provided by Sun Microsystems. In 11*g* Release 1 (11.1.1), all Java components and Java EE applications use the JKS-based keystore and truststore.

You use a JKS-based keystore for the following:

- Oracle Virtual Directory
- Applications deployed on Oracle WebLogic Server, including:
  - Oracle SOA Suite
  - Oracle WebCenter

In Oracle Fusion Middleware, you can use GUI or command-line tools to create, import, export, and delete a Java keystore and the certificates contained in the keystore. See Section 7.1.2, "Keystore Management Tools" for details.

While creating a keystore, you can pre-populate it with a keypair wrapped in a self-signed certificate; such a keystore is typically used in development and testing phases.

The other choice is to generate a certificate signing request for a keypair, 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 keystore; the keystore now contains a trusted certificate, since it comes from a trusted third-party. Such a keystore is typically used in production environments.

Keystores are always password-protected.

7.1.1.2 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.

You use an Oracle Wallet for the following components:

- Oracle HTTP Server
- Oracle Web Cache
- Oracle Internet Directory

In Oracle Fusion Middleware, you can use GUI or command-line tools to create, import, export and delete a wallet and the certificates contained in the wallet. See Section 7.1.2, "Keystore Management Tools" for details.

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.

The other choice is to 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.

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
- Oracle Wallet Manager, a stand-alone GUI tool for wallets, recommended for managing PKCS#11 wallets

This table shows the type of keystore used by each component, and the tool(s) available to manage the keystore:

Component/Application	Type of Keystore	Tasks	Tool
Oracle HTTP Server Oracle Web Cache Oracle Internet Directory	Oracle Wallet	Create Wallet, Create Certificate Request, Delete Wallet, Import Certificate, Export Certificate, Enable SSL	Fusion Middleware Control, WLST Oracle Wallet Manager and orapki for PKCS#11 or Hardware Security Modules (HSM)-based wallets. Also for environments where Fusion Middleware Control and WLST are not available (such as a stand-alone upgrade of these components without a domain).
Oracle Virtual Directory	JKS-based Keystore	Create KeyStore, Create Certificate Request, Delete KeyStore, Import Certificate, Export Certificate, Enable SSL	Fusion Middleware Control, WLST
Oracle SOA Suite	JKS-based Keystore	All Keystore operations	JDK Keytool
Oracle WebCenter	JKS-based Keystore	All Keystore operations	JDK Keytool
Oracle WebLogic Server	JKS-based Keystore	All Keystore operations	JDK Keytool
Oracle WebLogic Server	JKS-based Keystore	Enable SSL	Oracle WebLogic Server Administration Console
All Java EE applications (for example Oracle Directory Integration Platform, Oracle Directory Services Manager)	JKS-based Keystore	All Keystore operations	JDK Keytool

Note: Pre-11*g* wallets (corresponding to 10*g* Release 10.1.2 and 10.1.3 formats) are supported in 11*g* Release 1 (11.1.1).

About Importing DER-encoded Certificates

You cannot use Oracle Enterprise Manager Fusion Middleware Control or the `WLST` command-line tool to import DER-encoded certificates or trusted certificates into an Oracle wallet or a JKS keystore. Use these tools instead:

- To import DER-encoded certificates or trusted certificates into an Oracle wallet, use:
  - Oracle Wallet Manager or
  - `orapki` command-line tool
- To import DER-encoded certificates or trusted certificates into a JKS keystore, use the `keytool` utility.

Using a Keystore Not Created with WLST or Fusion Middleware Control

If an Oracle wallet or JKS keystore was created with tools such as `orapki` or `keytool`, it must be imported prior to use. Specifically:

- For Oracle HTTP Server, Oracle Web Cache, and Oracle Internet Directory, if a wallet was created using `orapki` or Oracle Wallet Manager, 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 Oracle Virtual Directory, if a keystore was created using `keytool`, in order to view or manage it in Fusion Middleware Control you must first import it with either Fusion Middleware Control or the WLST `importKeyStore` command.

Additional Information

Details about the tools are provided in these sections:

- Command-Line Interface for Keystores and Wallets
- JKS Keystore Management
- Wallet Management
- Appendix H, "Oracle Wallet Manager and orapki"

7.2 Command-Line Interface for Keystores and Wallets

Oracle Fusion Middleware provides a set of `wlst` scripts to create and manage JKS 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 3.5.1.1 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 JKS Keystore Management

This section describes the typical life cycle of keystores and certificates, and how to use Oracle Fusion Middleware tools to create and maintain keystores and certificates. It includes these topics:

- About Keystores and Certificates
- Managing the Keystore Life Cycle
- Common Keystore Operations
- Managing the Certificate Life Cycle
- Common Certificate Operations
- Keystore and Certificate Maintenance

7.3.1 About Keystores and Certificates

Keys and certificates are used to digitally sign and verify data and achieve authentication, integrity, and privacy in network communications.

A Java keystore (JKS) is a protected database that holds keys and certificates for the organization. Oracle Fusion Middleware utilizes JKS keystores for Oracle Virtual Directory and for applications deployed in Oracle WebLogic Server.

Access to a keystore requires a password which is defined at the time the keystore is created, by the person who creates the keystore, and which can only be changed by providing the current password.

In addition, each private key in a keystore can be secured by its own password.

This section contains these topics:

- Sharing Keystores Across Instances
- Keystore Naming Conventions

7.3.1.1 Sharing Keystores Across Instances

Oracle recommends that you do not share keystores between component instances or Oracle instances, since each keystore represents a unique identity.

The exception to this is an environment with a cluster of component instances, in which case keystore sharing would be an acceptable practice.

Note that no management tools or interfaces are available to facilitate keystore sharing. However, you can export a keystore from one instance and import it into another instance.

7.3.1.2 Keystore Naming Conventions

Follow these naming conventions for your JKS keystores:

- Do not use a name longer than 256 characters.
- Do not use any of the following characters in a keystore name:

  ```
  | ; , ! @ # $ ( ) < > / \ " ' ` ~ { } [ ] = + & ^ space tab
  ```

  Note: Observe this rule even if your operating system supports the character.

- Do not use non-ascii characters in a keystore name.
- Additionally, follow the operating system-specific rules for directory and file names.

7.3.2 Managing the Keystore Life Cycle

Typical life cycle events for a JKS keystore are as follows:

- The keystore is created. Keystores can be created directly, or by importing a keystore file from the file system.
- The list of available keystores are viewed and specific keystores selected for update.
- Keystores are updated or deleted. Update operations require that the keystore password be entered.
- The keystore password can be changed.
- The keystore can be deleted.
- Keystores can be exported and imported.

7.3.3 Common Keystore Operations

This section explains the following keystore operations:

- Creating a Keystore Using Fusion Middleware Control
- Creating a Keystore Using WLST
- Exporting a Keystore Using Fusion Middleware Control
- Exporting a Keystore Using WLST
- Deleting a Keystore Using Fusion Middleware Control
- Deleting a Keystore Using WLST
- Importing a Keystore Using Fusion Middleware Control
- Importing a Keystore Using WLST
- Changing the Keystore Password Using Fusion Middleware Control
- Changing the Keystore Password Using WLST

7.3.3.1 Creating a Keystore Using Fusion Middleware Control

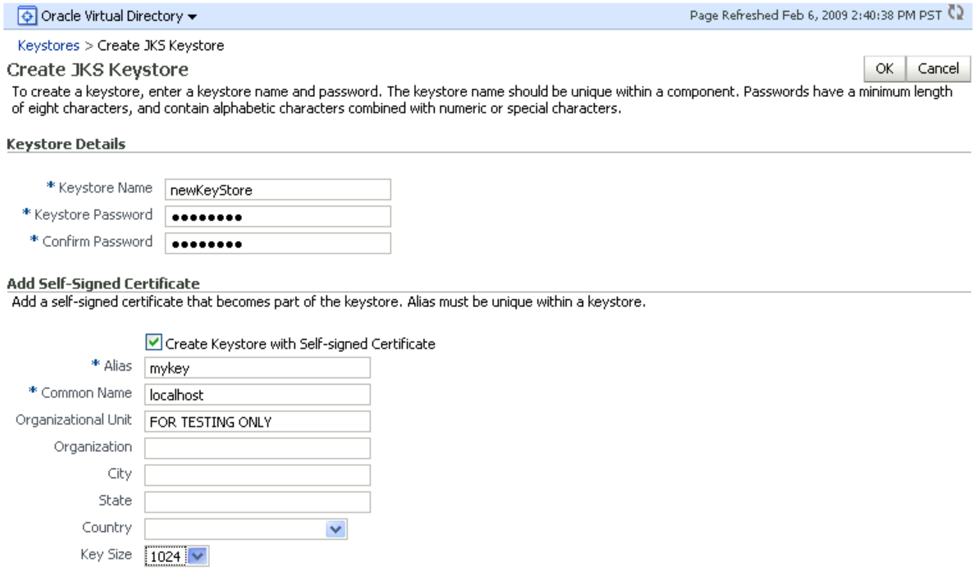
Take these steps to create a keystore:

1. Log in to the domain of interest using Fusion Middleware Control.
2. From the navigation pane, locate your component instance.
3. Navigate to *component_name*, then **Security**, then **Keystores**. For example, navigate to Oracle Virtual Directory, then **Security**, then **Keystores**.

> **Note:** The component type is displayed at the top of the page, adjacent to the Topology icon.

4. The Java Keystore page appears. On this page you can create, update, and delete keystores, and perform other keystore management tasks.
5. Click **Create**. The Create Keystore dialog appears.
6. Provide keystore details such as name and password.

   You can also request a self-signed certificate in this dialog, and fill in the alias name and DN information.



Oracle Virtual Directory ▾
Page Refreshed Feb 6, 2009 2:40:38 PM PST
Keystores > Create JKS Keystore
OK
Cancel

Create JKS Keystore

To create a keystore, enter a keystore name and password. The keystore name should be unique within a component. Passwords have a minimum length of eight characters, and contain alphabetic characters combined with numeric or special characters.

Keystore Details

* Keystore Name: newKeyStore
* Keystore Password: ••••••••
* Confirm Password: ••••••••

Add Self-Signed Certificate

Add a self-signed certificate that becomes part of the keystore. Alias must be unique within a keystore.

☑ Create Keystore with Self-signed Certificate

* Alias: mykey
* Common Name: localhost
Organizational Unit: FOR TESTING ONLY
Organization:
City:
State:
Country:
Key Size: 1024

Note: If you want to use this keystore only to store trusted certificates, you can uncheck the Create Self-Signed Certificate checkbox. This will create a keystore with no keypair.

---

7. Click **Submit**. The new keystore appears in the list of Java keystores.

7.3.3.2 Creating a Keystore Using WLST

Assuming the instance name is `inst1`, use this command to create a keystore:

```
createKeyStore('inst1', 'ovd5', 'ovd', 'newKeyStore', 'password')
```

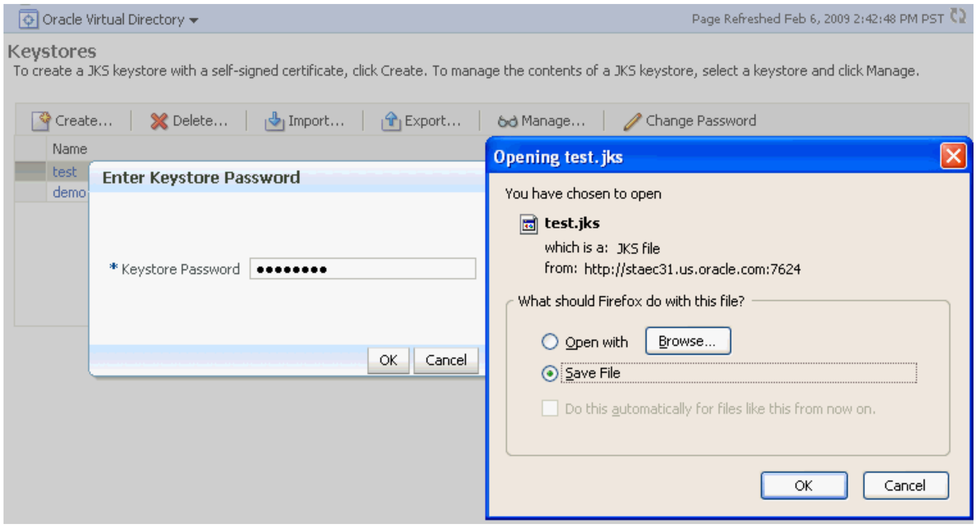
where `password` is the password for this keystore.

7.3.3.3 Exporting a Keystore Using Fusion Middleware Control

If multiple Oracle Virtual Directory instances want to share the same keystore file, this can be achieved by exporting the keystore from one instance and importing it into the other instances.

Take these steps to export a keystore:

1. Navigate to the Java Keystores page for the component instance, as explained in Section 7.3.3.1, "Creating a Keystore Using Fusion Middleware Control."
2. Select the desired keystore from the list of stores.
3. Click **Export**.
4. A dialog box appears in which you must enter the keystore password to continue.
5. Specify a file system location, and click **OK**.


Oracle Virtual Directory
Page Refreshed Feb 6, 2009 2:42:48 PM PST
Keystores
Create... | Delete... | Import... | Export... | Manage... | Change Password
Name
test
demo
OK | Cancel
Open with | Browse...
OK | Cancel

To create a JKS keystore with a self-signed certificate, click Create. To manage the contents of a JKS keystore, select a keystore and click Manage.

Enter Keystore Password

* Keystore Password ●●●●●●●●

Opening test.jks

You have chosen to open

test.jks

which is a: JKS file

from: http://staec31.us.oracle.com:7624

What should Firefox do with this file?

○ Open with

● Save File

□ Do this automatically for files like this from now on.

See Also: Section 7.3.3.7, "Importing a Keystore Using Fusion Middleware Control"

7.3.3.4 Exporting a Keystore Using WLST

Assuming the instance name is `inst1`, use this command to export a keystore:

```
exportKeyStore('inst1', 'ovd5', 'ovd', 'test', 'password', '/tmp')
```

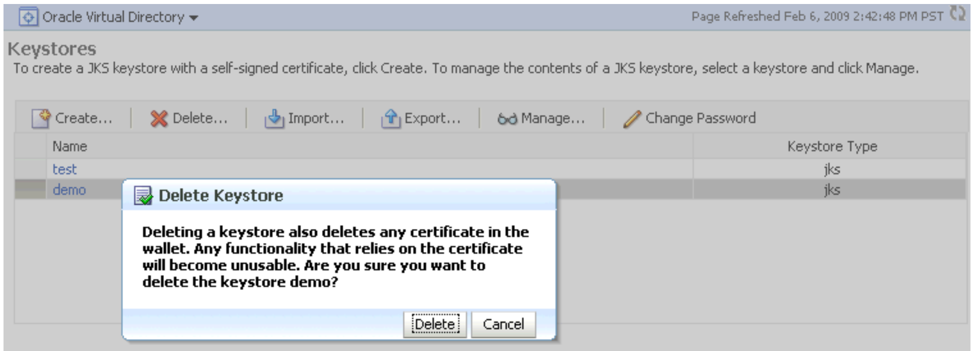
where `password` is the password for this keystore.

This command exports the keystore into a file named `test` under the directory `/tmp`.

7.3.3.5 Deleting a Keystore Using Fusion Middleware Control

Take these steps to delete a keystore:

1. Navigate to the Java Keystores page for the component instance, as explained in Section 7.3.3.1, "Creating a Keystore Using Fusion Middleware Control."
2. Select the desired keystore from the list of stores.
3. Click **Delete**.
4. A dialog box appears to request confirmation of the delete request.


Oracle Virtual Directory
Page Refreshed Feb 6, 2009 2:42:48 PM PST
Keystores
Create... | Delete... | Import... | Export... | Manage... | Change Password
Delete | Cancel

To create a JKS keystore with a self-signed certificate, click Create. To manage the contents of a JKS keystore, select a keystore and click Manage.

Name	Keystore Type
test	jks
demo	jks

Delete Keystore

Deleting a keystore also deletes any certificate in the wallet. Any functionality that relies on the certificate will become unusable. Are you sure you want to delete the keystore demo?

5. Click **Delete**.

7.3.3.6 Deleting a Keystore Using WLST

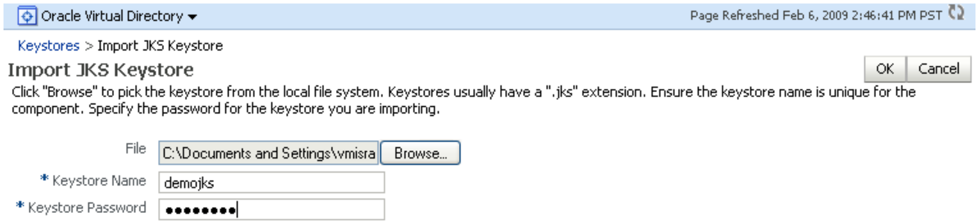
Assuming the application server instance name is `inst1`, use this command to delete a keystore:

```
deleteKeyStore('inst1', 'ovd5', 'ovd', 'demo')
```

where the component type is `ovd`, the component instance is `ovd5`, and the keystore is named `demo`.

7.3.3.7 Importing a Keystore Using Fusion Middleware Control

1. Navigate to the Java Keystores page for the component instance, as explained in Section 7.3.3.1, "Creating a Keystore Using Fusion Middleware Control."
2. Click **Import**.
3. The Import Keystore dialog box appears.
4. Browse the file system to locate the keystore file.
5. Provide a name for the keystore. Enter the keystore password.


Oracle Virtual Directory ▾
Page Refreshed Feb 6, 2009 2:46:41 PM PST
Keystores > Import JKS Keystore
OK
Cancel
Browse...

Import JKS Keystore

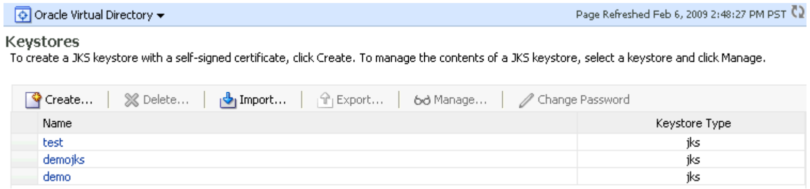
Click "Browse" to pick the keystore from the local file system. Keystores usually have a ".jks" extension. Ensure the keystore name is unique for the component. Specify the password for the keystore you are importing.

File: C:\Documents and Settings\vmisra

* Keystore Name: demojks

* Keystore Password: ••••••••

6. Click **OK**.
7. The imported keystore appears in the list of Java keystores.


Oracle Virtual Directory ▾
Page Refreshed Feb 6, 2009 2:48:27 PM PST
Create...
Delete...
Import...
Export...
Manage...
Change Password

Keystores

To create a JKS keystore with a self-signed certificate, click Create. To manage the contents of a JKS keystore, select a keystore and click Manage.

Name	Keystore Type
test	jks
demojks	jks
demo	jks

7.3.3.8 Importing a Keystore Using WLST

Assuming the instance name is inst1, use this command to import a keystore:

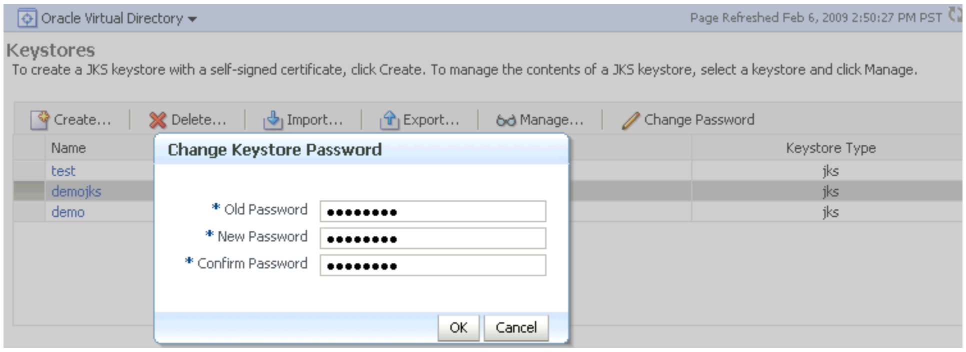
```
importKeyStore('inst1', 'ovd5', 'ovd', 'demojks', 'password', '/tmp/demojks.jks')
```

where `password` is the password for this keystore.

7.3.3.9 Changing the Keystore Password Using Fusion Middleware Control

Take these steps to change a keystore password:

1. Navigate to the Java Keystores page for the component instance, as explained in Section 7.3.3.1, "Creating a Keystore Using Fusion Middleware Control."
2. Select a keystore and click **Change Password**.
3. A dialog box appears on which you must enter the current password and enter a new password. The new password must be entered a second time to confirm.



Keystores

To create a JKS keystore with a self-signed certificate, click Create. To manage the contents of a JKS keystore, select a keystore and click Manage.

Name	Keystore Type
test	jks
demojks	jks
demo	jks

Change Keystore Password

* Old Password ••••••••

* New Password ••••••••

* Confirm Password ••••••••

4. Click **OK** to change the password. In future, any operations performed on this keystore or its certificates will require the use of the new password.

7.3.3.10 Changing the Keystore Password Using WLST

Assuming the instance name is `inst1`, use this command to change the keystore password:

```
changeKeyStorePassword('inst1', 'ovd5', 'ovd', 'demojks', 'current_password',
'new_password')
```

where `current_password` is the current password for this keystore, and `new_password` is the new password.

7.3.4 Managing the Certificate Life Cycle

Typical life cycle events for a certificate residing in a keystore are as follows:

- A self-signed certificate is automatically created for the keypair.
- A certificate signing request (CSR) is generated, and can then be exported to a file.
- Certificates are imported into the keystore. A certificate can either be pasted into a text box or imported from the file system. You can import both user certificates and trusted certificates (also known as CA certificates) in this way.
- Certificates or trusted certificates are exported from the keystore out to a file.
- Certificates or trusted certificates are deleted from the keystore.

7.3.5 Common Certificate Operations

This section describes the following common certificate operations:

- Generating a New Key for the Keystore Using Fusion Middleware Control
- Generating a New Key for the Keystore Using WLST
- Generating a Certificate Signing Request Using Fusion Middleware Control
- Generating a Certificate Signing Request Using WLST
- Importing a Certificate or Trusted Certificate into a Keystore Using Fusion Middleware Control
- Importing a Certificate or Trusted Certificate into a Keystore Using WLST
- Exporting a Certificate or Trusted Certificate from the Keystore Using Fusion Middleware Control
- Exporting a Certificate or Trusted Certificate from the Keystore Using WLST

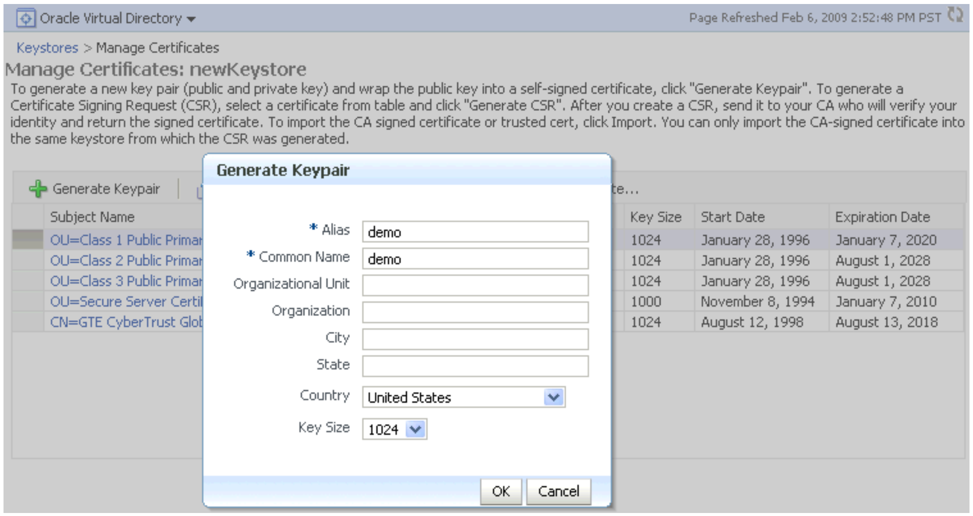
- Deleting a Certificate or Trusted Certificate from the Keystore Using Fusion Middleware Control
- Deleting a Certificate or Trusted Certificate from the Keystore Using WLST
- Converting a Self-Signed Certificate to a Third-Party Certificate Using Fusion Middleware Control
- Converting a Self-Signed Certificate to a Third-Party Certificate Using WLST

7.3.5.1 Generating a New Key for the Keystore Using Fusion Middleware Control

To generate a new key (that is, a new self-signed certificate) for a keystore:

1. Navigate to the Java Keystores page for the component instance, as explained in Section 7.3.3.1, "Creating a Keystore Using Fusion Middleware Control."
2. Select the keystore from the list of stores.
3. A dialog box appears in which you must enter the keystore password to continue.
4. The Manage Certificates page appears. Here, you can manage both types of keystore entries, that is, certificates and trusted certificates.
5. Click the **Generate Keypair** button.
6. In the Generate Keypair dialog, enter the details for the new key and click **OK**.

Example: Generating a Key Pair



Page Refreshed Feb 6, 2009 2:52:48 PM PST

Manage Certificates: newKeystore

To generate a new key pair (public and private key) and wrap the public key into a self-signed certificate, click "Generate Keypair". To generate a Certificate Signing Request (CSR), select a certificate from table and click "Generate CSR". After you create a CSR, send it to your CA who will verify your identity and return the signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CA-signed certificate into the same keystore from which the CSR was generated.

Subject Name	Key Size	Start Date	Expiration Date
OU=Class 1 Public Primar	1024	January 28, 1996	January 7, 2020
OU=Class 2 Public Primar	1024	January 28, 1996	August 1, 2028
OU=Class 3 Public Primar	1024	January 28, 1996	August 1, 2028
OU=Secure Server Certil	1000	November 8, 1994	January 7, 2010
CN=GTE CyberTrust Glot	1024	August 12, 1998	August 13, 2018

Generate Keypair

* Alias: demo
* Common Name: demo
Organizational Unit:
Organization:
City:
State:
Country: United States
Key Size: 1024

When you complete these steps, a new public-private key pair is generated for the keystore, and the public key is wrapped in a self-signed certificate.

While these steps generate a new keypair for an existing keystore, you can also generate a new keypair when creating the keystore itself. For details, see Section 7.3.3.1, "Creating a Keystore Using Fusion Middleware Control."

7.3.5.2 Generating a New Key for the Keystore Using WLST

Assuming the instance name is `inst1`, use this command to generate a new key for a keystore:

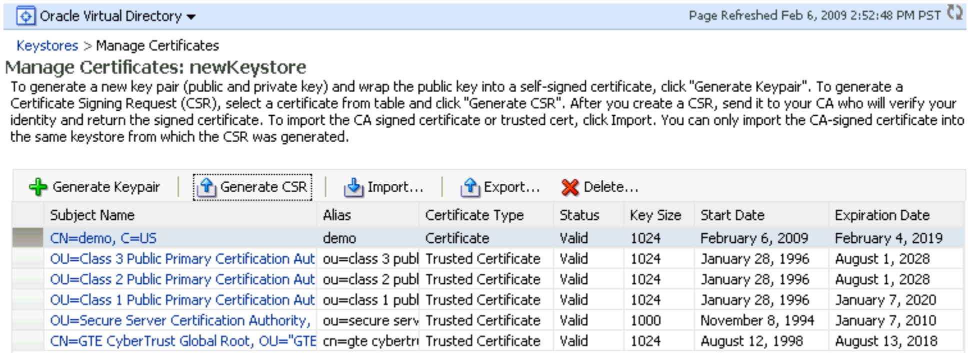
```
generateKey('inst1', 'ovd5', 'ovd', 'newKeystore', 'password', 'subject_dn', 'key_
size', 'alias')
```

where `password` is the password for this keystore, `subject_dn` is the distinguished name by which the key pair is generated, `key_size` is the key size in bits, and `alias` is the key alias.

7.3.5.3 Generating a Certificate Signing Request Using Fusion Middleware Control

Take these steps to create a Certificate Signing Request (CSR):

1. From the navigation pane, locate your component instance.
2. Navigate to *component_name*, then **Security**, then **Keystores**.
3. Select the desired keystore from the list of stores.
4. A dialog box appears in which you must enter the keystore password to continue.
5. The Manage Certificates page appears. Select the self-signed certificate for which you want to generate the CSR and click **Generate CSR**.

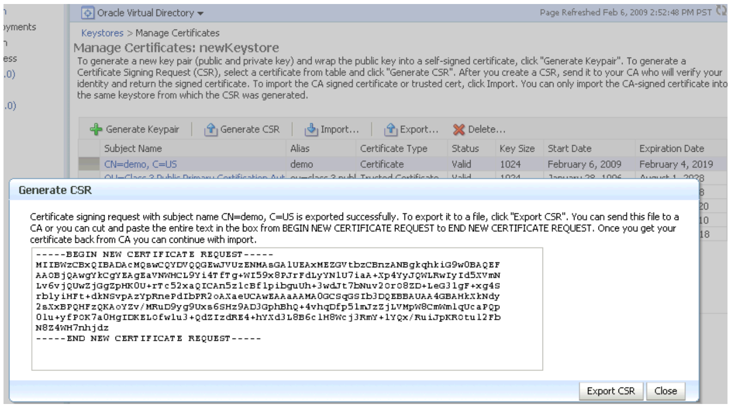


Manage Certificates: newKeystore

To generate a new key pair (public and private key) and wrap the public key into a self-signed certificate, click "Generate Keypair". To generate a Certificate Signing Request (CSR), select a certificate from table and click "Generate CSR". After you create a CSR, send it to your CA who will verify your identity and return the signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CA-signed certificate into the same keystore from which the CSR was generated.

Subject Name	Alias	Certificate Type	Status	Key Size	Start Date	Expiration Date
CN=demo, C=US	demo	Certificate	Valid	1024	February 6, 2009	February 4, 2019
OU=Class 3 Public Primary Certification Aut	ou=class 3 publ	Trusted Certificate	Valid	1024	January 28, 1996	August 1, 2028
OU=Class 2 Public Primary Certification Aut	ou=class 2 publ	Trusted Certificate	Valid	1024	January 28, 1996	August 1, 2028
OU=Class 1 Public Primary Certification Aut	ou=class 1 publ	Trusted Certificate	Valid	1024	January 28, 1996	January 7, 2020
OU=Secure Server Certification Authority,	ou=secure serv	Trusted Certificate	Valid	1000	November 8, 1994	January 7, 2010
CN=GTE CyberTrust Global Root, OU="GTE	cn=gte cybertr	Trusted Certificate	Valid	1024	August 12, 1998	August 13, 2018

6. A dialog box appears, showing the generated signing request. You can either:
   - Copy the CSR from the dialog box and past it to a file.
   - Click the **Export CSR** button to directly save it to a file.


Oracle Virtual Directory ▾
Page Refreshed Feb 6, 2009 2:52:48 PM PST
Keystores > Manage Certificates
Manage Certificates: newKeystore
To generate a new key pair (public and private key) and wrap the public key into a self-signed certificate, click "Generate Keypair". To generate a Certificate Signing Request (CSR), select a certificate from table and click "Generate CSR". After you create a CSR, send it to your CA who will verify your identity and return the signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CA-signed certificate into the same keystore from which the CSR was generated.

Generate CSR

Certificate signing request with subject name CN=demo, C=US is exported successfully. To export it to a file, click "Export CSR". You can send this file to a CA or you can cut and paste the entire text in the box from BEGIN NEW CERTIFICATE REQUEST to END NEW CERTIFICATE REQUEST. Once you get your certificate back from CA you can continue with import.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBWzCBxQIBADAcMQswCQYDVQQGEwJVUzENMAsGA1UEAxMEZGVtbzCBnzANBgkqhkiG9w0BAQEF
AAOBjQAwgYkCgYEAgEaVNWHCL9Yi4TfTg+WI59x8PJrFdLyYN1U7iaA+Xp4YyJQWLRwIyId5XVmN
Lv6vjQUwZjGgZpHK0U+rTc52xaQICAn5z1cBf1pibguUh+3wdJt7bNuv2Or08ZD+LeG31gF+xg4S
rblyiHFt+dkNSvpAzYpRnePdIbPR2oAXaeUCAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBAHkXkNdy
2sXxBPQHFzQKAoYZv/MRuD9yg9Uxs6SHz9AD3GphBhQ+4vhqDfp51mJzZjLVMpW8CmWmlqUcaPQp
Olu+yfPOK7a0HgIDKELOfw1u3+QdZIzdRE4+hYXd3L8B6c1H8Wcj3RmY+1YQx/RuiJpKROtul2Fb
N8Z4WH7nhjdz
-----END NEW CERTIFICATE REQUEST-----
```

7.3.5.4 Generating a Certificate Signing Request Using WLST

Assuming the instance name is `inst1`, use this command to generate and export a CSR:

```
exportKeyStoreObject('inst1', 'ovd5', 'ovd', 'newKeystore', 'password',
'CertificateRequest', '/tmp', 'alias')
```

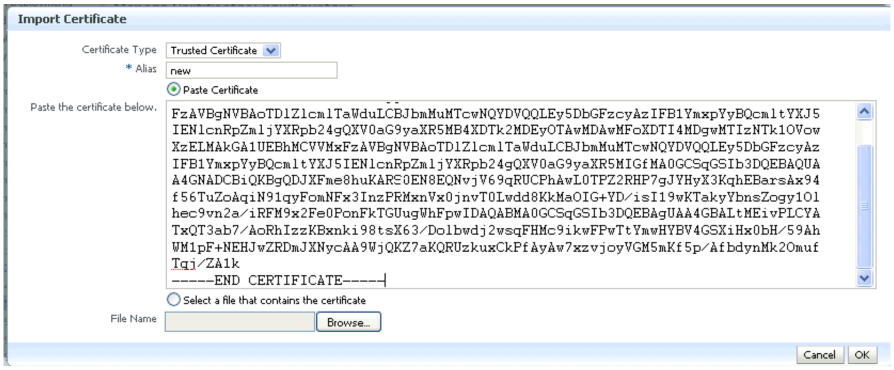
where `password` is the password for this keystore, `/tmp` is the path under which the certificate request is generated in BASE64 format in the file `base64.txt`, and `alias` is the alias of the key pair that is used to generate the certificate request.

7.3.5.5 Importing a Certificate or Trusted Certificate into a Keystore Using Fusion Middleware Control

> **Note:** You cannot use Fusion Middleware Control to import DER-encoded certificates or trusted certificates into a JKS keystore; use the `keytool` utility for this task.

Take these steps to import a certificate, or a trusted certificate, into a keystore:

1. From the navigation pane, locate your component instance.
2. Navigate to *component_name*, then **Security**, then **Keystores**.
3. Select the desired keystore from the list of stores.
4. A dialog box appears in which you must enter the keystore password to continue.
5. The Manage Certificates page appears. Click the **Import** button.
6. A dialog box appears with which you can either:
   - Paste the Base-64 encoded contents of a certificate or trusted certificate into the keystore directly.
   - Select a certificate or trusted certificate file from the file system.

   Import Certificate

   Certificate Type: Trusted Certificate

   * Alias: new

   ● Paste Certificate

   Paste the certificate below.

```
FzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAzIFB1YmxpYyBQcmltYXJ5
IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTk2MDEyOTAwMDAwMFoXDTI4MDgwMTIzNTk1OVow
XzELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAz
IFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDJXFme8huKARS0EN8EQNvjV69qRUCPhAwL0TPZ2RHP7gJYHyX3KqhEBarsAx94
f56TuZoAqiN91qyFomNFx3InzPRMxnVx0jnvT0Lwdd8KkMaOIG+YD/isI19wKTakyYbnsZogy1Ol
hec9vn2a/iRFM9x2Fe0PonFkTGUugWhFpwIDAQABMA0GCSqGSIb3DQEBAgUAA4GBALtMEivPLCYA
TxQT3ab7/AoRhIzzKBxnki98tsX63/Dolbwdj2wsqFHMc9ikwFPwTtYmwHYBV4GSXiHx0bH/59Ah
WM1pF+NEHJwZRDmJXNycAA9WjQKZ7aKQRUzkuxCkPfAyAw7xzvjoyVGM5mKf5p/AfbdynMk2Omuf
Tqj/ZA1k
-----END CERTIFICATE-----
```

   ○ Select a file that contains the certificate

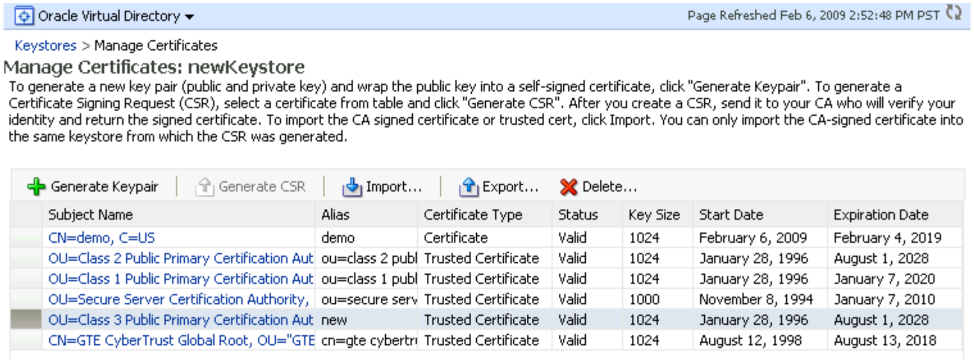
   File Name:

   You need to specify an alias while importing a certificate.

   When importing a certificate, the alias should match the alias of the corresponding keypair.

   When importing a trusted certificate, the alias should be unique in the keystore.

7. Click **OK**. The Manage Certificates page appears, showing the newly imported certificate or trusted certificate.



Oracle Virtual Directory
Page Refreshed Feb 6, 2009 2:52:48 PM PST
Keystores > Manage Certificates

Manage Certificates: newKeystore

To generate a new key pair (public and private key) and wrap the public key into a self-signed certificate, click "Generate Keypair". To generate a Certificate Signing Request (CSR), select a certificate from table and click "Generate CSR". After you create a CSR, send it to your CA who will verify your identity and return the signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CA-signed certificate into the same keystore from which the CSR was generated.

Subject Name	Alias	Certificate Type	Status	Key Size	Start Date	Expiration Date
CN=demo, C=US	demo	Certificate	Valid	1024	February 6, 2009	February 4, 2019
OU=Class 2 Public Primary Certification Aut	ou=class 2 publ	Trusted Certificate	Valid	1024	January 28, 1996	August 1, 2028
OU=Class 1 Public Primary Certification Aut	ou=class 1 publ	Trusted Certificate	Valid	1024	January 28, 1996	January 7, 2020
OU=Secure Server Certification Authority,	ou=secure serv	Trusted Certificate	Valid	1000	November 8, 1994	January 7, 2010
OU=Class 3 Public Primary Certification Aut	new	Trusted Certificate	Valid	1024	January 28, 1996	August 1, 2028
CN=GTE CyberTrust Global Root, OU="GTE	cn=gte cybertr	Trusted Certificate	Valid	1024	August 12, 1998	August 13, 2018

7.3.5.6 Importing a Certificate or Trusted Certificate into a Keystore Using WLST

> **Note:** You cannot use the `WLST` command-line tool to import DER-encoded certificates or trusted certificates into a JKS keystore; use the `keytool` utility for this purpose.

Assuming the instance name is `inst1`, use this command to import a certificate into a keystore:

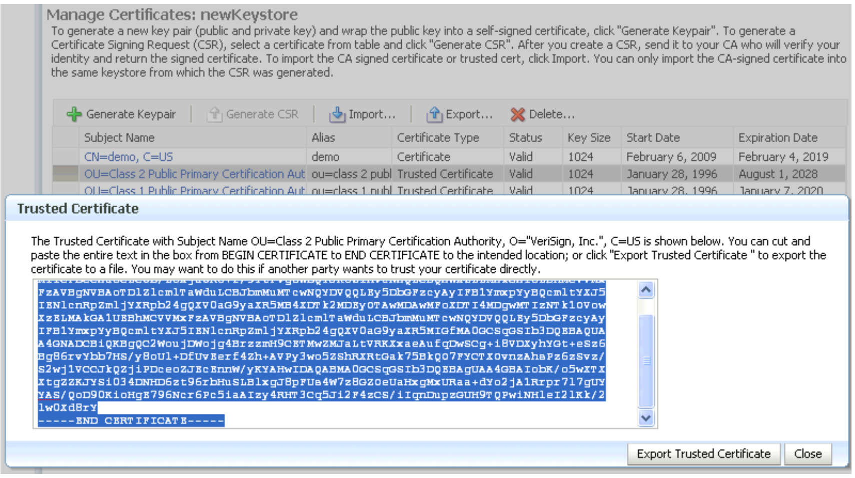
```
importKeyStoreObject('inst1', 'ovd5', 'ovd', 'newKeystore', 'password',
'Certificate', '/tmp/cert.txt', 'alias')
```

where `password` is the password for this keystore, `/tmp/cert.txt` is the file that contains BASE64 encoded certificate, and `alias` is the alias by which this certificate is imported. Note that this alias must be same as that of the key pair that was used to generate this certificate request.

7.3.5.7 Exporting a Certificate or Trusted Certificate from the Keystore Using Fusion Middleware Control

Take these steps to export a certificate or trusted certificate from the keystore:

1. From the navigation pane, locate your component instance.
2. Navigate to *component_name*, then **Security**, then **Keystores**.
3. Select the desired keystore from the list of stores.
4. A dialog box appears in which you must enter the keystore password to continue.
5. The Manage Certificates page appears. Click **Export**.
6. A dialog box appears which shows the Base64 encoded certificate or trusted certificate. You can either copy the contents from the text box and paste it to a file, or select the **Export** button to save it directly to a file.



Manage Certificates: newKeystore

To generate a new key pair (public and private key) and wrap the public key into a self-signed certificate, click "Generate Keypair". To generate a Certificate Signing Request (CSR), select a certificate from table and click "Generate CSR". After you create a CSR, send it to your CA who will verify your identity and return the signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CA-signed certificate into the same keystore from which the CSR was generated.

Subject Name	Alias	Certificate Type	Status	Key Size	Start Date	Expiration Date
CN=demo, C=US	demo	Certificate	Valid	1024	February 6, 2009	February 4, 2019
OU=Class 2 Public Primary Certification Aut	ou=class 2 publ	Trusted Certificate	Valid	1024	January 28, 1996	August 1, 2028
OU=Class 1 Public Primary Certification Aut	ou=class 1 publ	Trusted Certificate	Valid	1024	January 28, 1996	January 7, 2020

Trusted Certificate

The Trusted Certificate with Subject Name OU=Class 2 Public Primary Certification Authority, O="VeriSign, Inc.", C=US is shown below. You can cut and paste the entire text in the box from BEGIN CERTIFICATE to END CERTIFICATE to the intended location; or click "Export Trusted Certificate " to export the certificate to a file. You may want to do this if another party wants to trust your certificate directly.

```
FzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAyIFB1YmxpYyBQcmltYXJ5
IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTk2MDEyOTAwMDAwMFoXDTI4MDgwMTIzNTk1OVow
XzELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAy
IFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQC2WoujDWojg4BrzzmH9CETMwZMJaLtVRKXxaeAufqDwSCg+i8VDXyhYGt+eSz6
Bg86rvYbb7HS/y8oUl+DfUvEerf4Zh+AVPy3wo5ZShRXRtGak75BkQO7FYCTXOvnzAhePz6zSvz/
S2wj1VCCJkQZjiPDceoZJEcEnnW/yKYAHwIDAQABMA0GCSqGSIb3DQEBAgUAA4GBAIobK/o5wXTX
XtgZZKJYSi034DNHD6zt96rbHuSLBlxgJ8pFUs4W7z8GZOeUaHxgMxURaa+dYo2jA1Rrpr717gUY
YAS/QoD90KioHgE796Ncr6Pc5iaAIzy4RHT3Cq5Ji2F4zCS/iIqnDupzGUH9TQPwiNHleI2lKk/2
lwOxd8rY
-----END CERTIFICATE-----
```

7.3.5.8 Exporting a Certificate or Trusted Certificate from the Keystore Using WLST

Assuming the instance name is `inst1`, use this command to export a certificate:

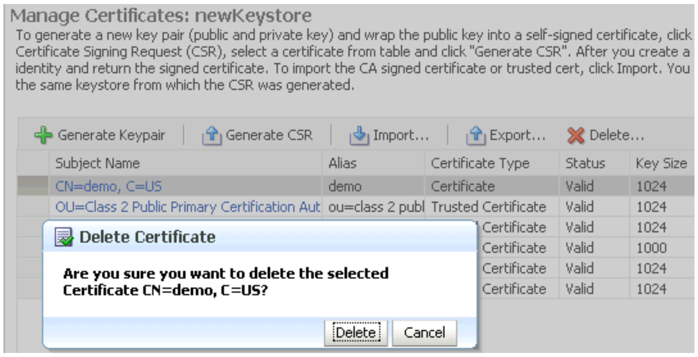
```
exportKeyStoreObject('inst1', 'ovd5', 'ovd', 'newKeystore', 'password',
'Certificate', '/tmp', 'alias')
```

where `password` is the password for this keystore, `/tmp` is the path under which the certificate is generated in BASE64 format in the file `base64.txt`, and `alias` is the alias of the certificate being exported.

7.3.5.9 Deleting a Certificate or Trusted Certificate from the Keystore Using Fusion Middleware Control

Take these steps to delete a certificate, or a trusted certificate, from a keystore:

1. From the navigation pane, locate your component instance.
2. Navigate to *component_name*, then **Security**, then **Keystores**.
3. Select the desired keystore from the list of stores.
4. A dialog box appears in which you must enter the keystore password to continue.
5. The Manage Certificates page appears. Select the certificate or trusted certificate to be deleted, and Click **Delete**.
6. A dialog box appears asking you to confirm the choice. Select **OK** to confirm.



Manage Certificates: newKeystore

To generate a new key pair (public and private key) and wrap the public key into a self-signed certificate, click
Certificate Signing Request (CSR), select a certificate from table and click "Generate CSR". After you create a
identity and return the signed certificate. To import the CA signed certificate or trusted cert, click Import. You
the same keystore from which the CSR was generated.

Subject Name	Alias	Certificate Type	Status	Key Size
CN=demo, C=US	demo	Certificate	Valid	1024
OU=Class 2 Public Primary Certification Aut	ou=class 2 publ	Trusted Certificate	Valid	1024
		Certificate	Valid	1024
		Certificate	Valid	1000
		Certificate	Valid	1024
		Certificate	Valid	1024

Delete Certificate

Are you sure you want to delete the selected Certificate CN=demo, C=US?

7.3.5.10 Deleting a Certificate or Trusted Certificate from the Keystore Using WLST

Assuming the application server instance name is `inst1`, use this command to delete a certificate:

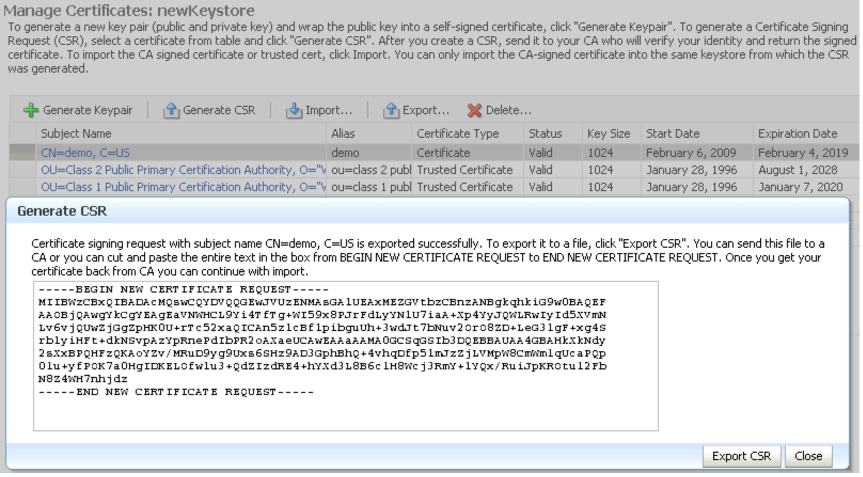
```
removeKeyStoreObject('inst1', 'ovd5', 'ovd', 'newKeystore', 'password',
'Certificate', 'alias')
```

where `password` is the password for this keystore and `alias` is the alias of the certificate being deleted.

7.3.5.11 Converting a Self-Signed Certificate to a Third-Party Certificate Using Fusion Middleware Control

Take these steps to convert a self-signed certificate, residing in a keystore, into a third-party certificate:

1. From the navigation pane, locate your component instance.
2. Navigate to *component_name*, then **Security**, then **Keystores**.
3. Select the keystore that contains the self-signed certificate from the list of stores.
4. A dialog box appears in which you must enter the keystore password; click **OK** to continue.
5. The Manage Certificates page appears.
6. A new certificate request must be generated for the self-signed certificate that is to be converted. Select the certificate and click **Generate CSR**. In this example, the request is made for the self-signed certificate with alias `demo`.



Manage Certificates: newKeystore

To generate a new key pair (public and private key) and wrap the public key into a self-signed certificate, click "Generate Keypair". To generate a Certificate Signing Request (CSR), select a certificate from table and click "Generate CSR". After you create a CSR, send it to your CA who will verify your identity and return the signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CA-signed certificate into the same keystore from which the CSR was generated.

Subject Name	Alias	Certificate Type	Status	Key Size	Start Date	Expiration Date
CN=demo, C=US	demo	Certificate	Valid	1024	February 6, 2009	February 4, 2019
OU=Class 2 Public Primary Certification Authority, O="V	ou=class 2 publ	Trusted Certificate	Valid	1024	January 28, 1996	August 1, 2028
OU=Class 1 Public Primary Certification Authority, O="V	ou=class 1 publ	Trusted Certificate	Valid	1024	January 28, 1996	January 7, 2020

Generate CSR

Certificate signing request with subject name CN=demo, C=US is exported successfully. To export it to a file, click "Export CSR". You can send this file to a CA or you can cut and paste the entire text in the box from BEGIN NEW CERTIFICATE REQUEST to END NEW CERTIFICATE REQUEST. Once you get your certificate back from CA you can continue with import.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBWzCBxQIBADAcMQswCQYDVQQGEwJVUzENMAsGA1UEAxMEZGVtbzCBnzANBgkqhkiG9w0BAQEF
AAOBjQAwgYkCgYEAgEaVNWHCL9Yi4TfTg+WI59x8PJrFdLyYN1U7iaA+Xp4YyJQWLRwIyId5XVmN
Lv6vjQUwZjGgZpHK0U+rTc52xaQICAn5z1cBf1pibguUh+3wdJt7bNuv2Or08ZD+LeG31gF+xg4S
rb1yiHFt+dkNSvpAzYpRnePdIbPR2oAXaeUCAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBAHkXkNdy
2sXxBPQHFzQKAoYZv/MRuD9yg9Uxs6SHz9AD3GphBhQ+4vhqDfp51mJzZjLVMpW8CmWm1qUcaPQp
01u+yfPOK7a0HgIDKEL0fw1u3+QdZIzdRE4+hYXd3L8B6c1H8Wcj3RmY+1YQx/RuiJpKROtu12Fb
N8Z4WH7nhjdz
-----END NEW CERTIFICATE REQUEST-----
```

The certificate request is displayed.

7. You can either:
   - Copy and paste the Base64-encoded certificate request to a file.
   - Export it directly to a file with the **Export CSR** button.
8. Submit the certificate request file to a certificate authority (CA).
9. 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

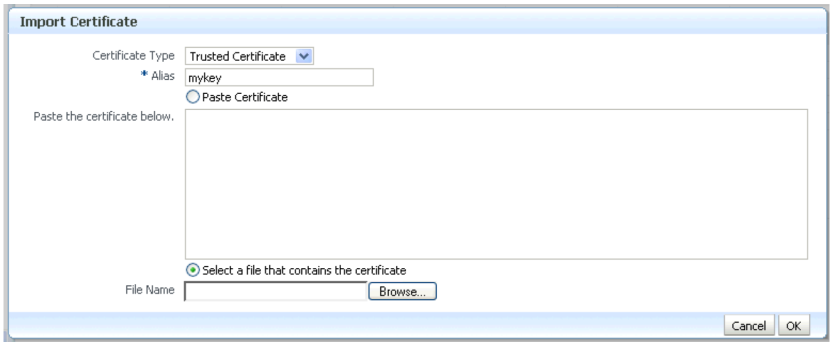
- Two files, one containing the newly generated certificate and a second containing its own CA certificate

10. Use **Import** to import these files into your keystore:

    - If you received a single file from the CA, import it as a certificate, using an alias that matches the alias of the self-signed certificate you are replacing (from Step 6)
    - If you received two files:
      – Import the file containing the CA certificate as a trusted certificate (use an alias that is unique in the keystore)
      – 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.

    The CA returned a single file, which is imported as a certificate:



Import Certificate

Certificate Type: Trusted Certificate

* Alias: mykey

○ Paste Certificate

Paste the certificate below.

● Select a file that contains the certificate

File Name

11. After import, the certificate issued by the CA replaces the self-signed certificate.

7.3.5.12 Converting a Self-Signed Certificate to a Third-Party Certificate Using WLST

Use these steps to convert a self-signed certificate to a third-party certificate:

1. Generate and export a CSR.

```
exportKeyStoreObject('inst1', 'ovd5', 'ovd', 'jks1', '<password>',
'CertificateRequest', '/tmp', 'mykey')
```

2. Submit the CSR `/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 two separate files.

3. If you receive a single file from the CA, run the command:

```
importKeyStoreObject('inst1', 'ovd5', 'ovd', 'jks1', 'password', 'Certificate',
'/tmp/cert.txt', 'alias')
```

   where `password` is the password for this keystore, `/tmp/cert.txt` is the file that the CA returned and contains the BASE64 encoded PKCS#7, and `alias` is the alias by which this certificate is imported. Note that this alias must match that of the key pair that was used to generate the certificate request.

If you receive two files from the CA, import the CA certificate first as a trusted certificate, followed by the newly generated certificate:

```
importKeyStoreObject('inst1', 'ovd5', 'ovd', 'jks1', 'password',
'TrustedCertificate', '/tmp/cacert.txt', 'unique_alias')
```

where `unique_alias` is a unique alias by which the trusted certificate is imported.

```
importKeyStoreObject('inst1', 'ovd5', 'ovd', 'jks1', 'password', 'Certificate',
'/tmp/cert.txt', 'alias')
```

where `password` is the password for this keystore, `/tmp/cert.txt` is the file that the CA returned and contains BASE64 encoded certificate, `/tmp/cacert.txt` is the file containing the BASE64 encoded CA certificate, and `alias` is the alias by which this certificate is imported. Note that this alias must match that of the key pair that was used to generate the certificate request.

7.3.6 Keystore and Certificate Maintenance

This section contains the following administration topics:

- Location of Keystores
- Replacing Expiring Certificates
- Effect of Host Name Change on Keystores

7.3.6.1 Location of Keystores

The root directory for Oracle Virtual Directory keystores is located in `$ORACLE_INSTANCE/config/OVD/`*`ovd_instance_name`*`/keystores`.

This root directory will contain all the JKS files.

A sample structure, assuming there are two keystores named `keys.jks` and `trust.jks` respectively, would look like this:

```
ORACLE_INSTANCE/config/OVD/ovd_instance_name/keystores/keys.jks
ORACLE_INSTANCE/config/OVD/ovd_instance_name/keystores/trust.jks
```

7.3.6.2 Replacing Expiring Certificates

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. Generate a certificate request from the keystore (use the same key-pair 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 ask the CA to issue a new certificate for that certificate request.

3. Import the newly issued certificate into the keystore using the same alias as that of the key-pair.
4. 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.

7.3.6.3 Effect of Host Name Change on Keystores

Typically, the certificate DN is based on the host name of the server where the keystore is used.

For example, if a keystore is being created for the Oracle Virtual Directory server on host my.example.com, then the DN of the certificate in this Oracle Virtual Directory keystore 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 HTTP Client, among others. If the host name of the server does not match that of the certificate DN:

- A clear warning is displayed (in the case of browser clients).
- There may be SSL handshake failure (in the case of other clients).

Thus, whenever you have a keystore 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 keystore.

This can be done by requesting a new certificate with a new DN (based on the new host name).

For a Production Keystore

The steps are:

1. Generate a new request with the new DN (based on a new host name).
2. Send this request to a certificate authority (CA).
3. Get back a new certificate from the CA.
4. Import the new certificate with the same alias as the key-pair for which certificate request was generated.

For a Self-signed Keystore

The steps are:

1. Delete the existing keystore.
2. Create a new keystore with a key-pair using the new DN (based on the new host name).

For Both Keystore Types

For both production and self-signed keystores, once the new certificate is available in the keystore, ensure that it is imported into all the component keystores where it needs to be trusted. For example, if the HTTP listener on Oracle Virtual Directory was SSL-enabled and its certificate changed due to a host name change, then you need to import its new certificate into the client keystore or browser repository so that it can trust its new peer.

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 11*g* Release 1 (11.1.1), 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 10*g* 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.

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 or Oracle 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.

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).
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:

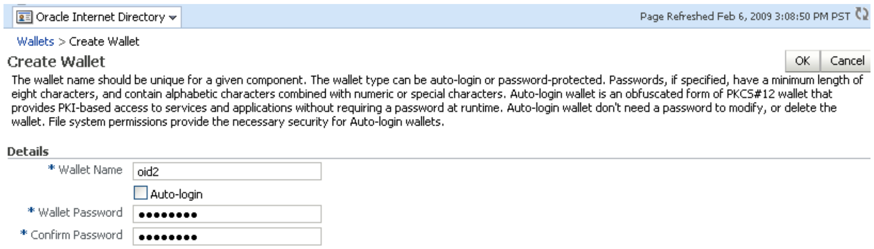
- Creating a Wallet Using Fusion Middleware Control
- Creating a Wallet Using WLST
- Creating a Self-Signed Wallet Using Fusion Middleware Control
- Creating a Self-Signed Wallet Using WLST
- Changing a Self-Signed Wallet to a Third-Party Wallet Using Fusion Middleware Control
- Changing a Self-Signed 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.


Oracle Internet Directory
Page Refreshed Feb 6, 2009 3:08:50 PM PST
Wallets > Create Wallet
OK
Cancel

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: oid2

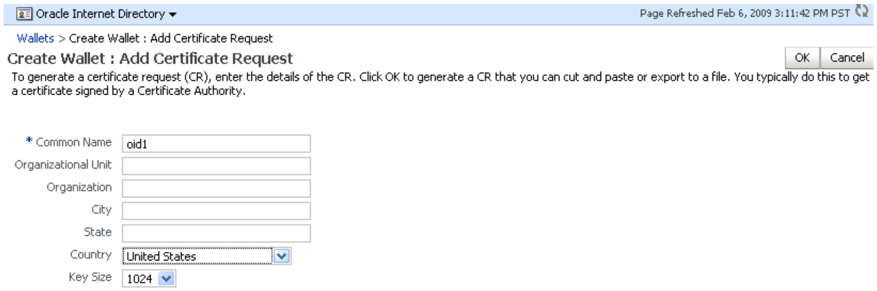
□ Auto-login

* 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
OK
Cancel

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	oid1
Organizational Unit	
Organization	
City	
State	
Country	United States
Key Size	1024

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

> **Note:** The WLST commands described in this chapter use Oracle Internet Directory as the example component. The same commands can be executed for Oracle HTTP Server or Oracle Web Cache by changing the third parameter from "`oid`" to "`ohs`" or "`webcache`" respectively.

Assuming the instance name is `inst1`, use this command to create a wallet:

```
createWallet('inst1', 'oid1', 'oid', 'oid2', 'password')
```

where `oid2` 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).

7.4.4.3 Creating a Self-Signed Wallet Using Fusion Middleware Control

Take these steps to create a self-signed wallet:

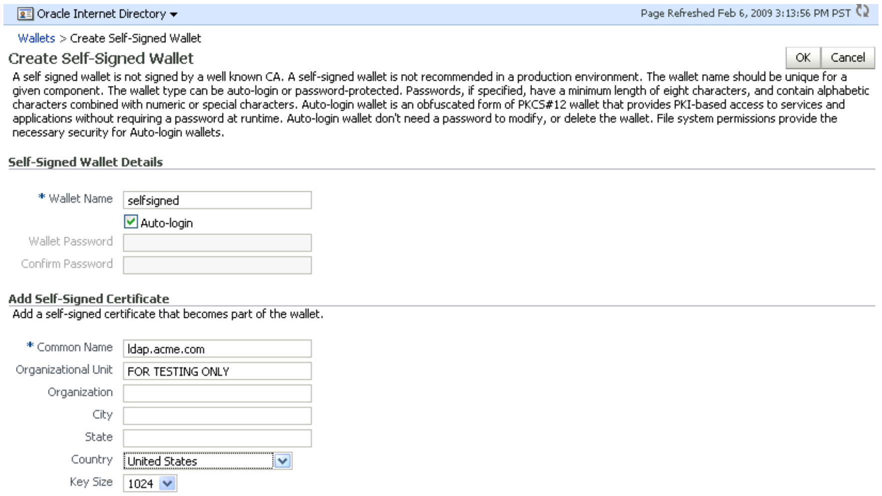
> **See Also:** Section 7.4.1.2, "Self-Signed and Third-Party Wallets"

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
OK
Cancel

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:

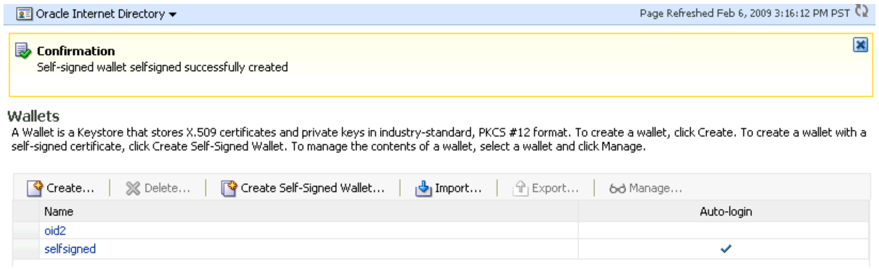
State:

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
Create...
Delete...
Create Self-Signed Wallet...
Import...
Export...
Manage...

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	✓

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', 'oid1', 'oid', 'oid2', 'password')
addSelfSignedCertificate('inst1', 'oid1', 'oid', 'oid2', 'password', 'subject_dn',
'key_size')
```

where `oid2` 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).

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."

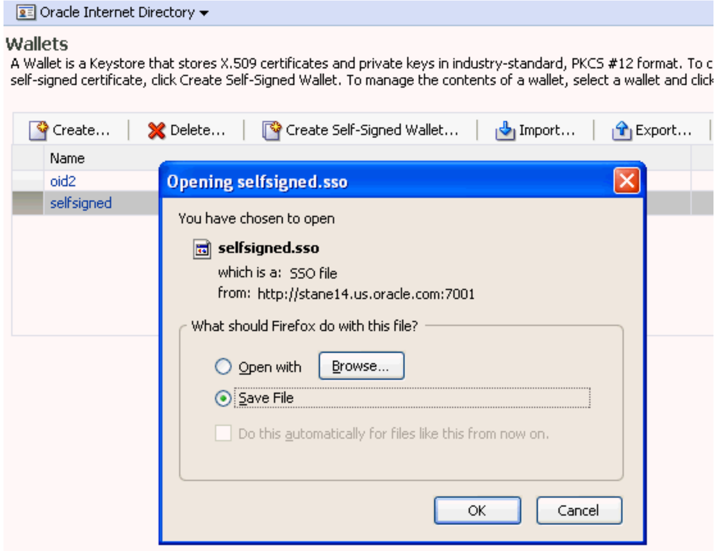
7.4.4.7 Exporting a Wallet Using Fusion Middleware Control

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**.



A Wallet is a Keystore that stores X.509 certificates and private keys in industry-standard, PKCS #12 format. To c
self-signed certificate, click Create Self-Signed Wallet. To manage the contents of a wallet, select a wallet and click

You have chosen to open

selfsigned.sso

which is a: SSO file

from: http://stane14.us.oracle.com:7001

What should Firefox do with this file?

○ Open with Browse...

● Save File

□ Do this automatically for files like this from now on.

OK Cancel

7.4.4.8 Exporting a Wallet Using WLST

Assuming the instance name is `inst1`, use this command to export a wallet:

```
exportWallet('inst1', 'oid1', 'oid', 'selfsigned', 'password', '/tmp')
```

where `password` is the password for this wallet (specify '' as password for auto-login wallet).

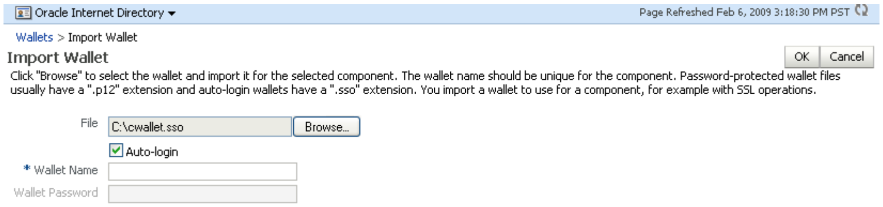
If it is an auto-login wallet, this command will export the wallet into a file named cwallet.sso under the directory `/tmp`. If it is a password-protected wallet, there will be two files created under `/tmp`, namely `ewallet.p12` and `cwallet.sso`.

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
OK
Cancel
Browse...

Import Wallet

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.

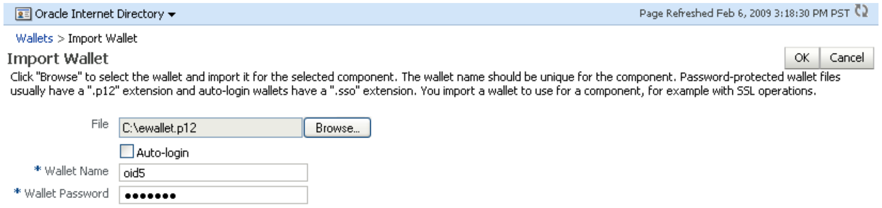
File: C:\cwallet.sso

☑ 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
OK
Cancel
Browse...

Import Wallet

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

□ Auto-login

* Wallet Name: oid5

* 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:

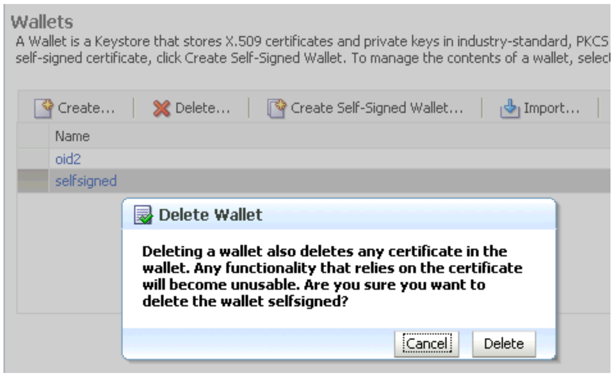
```
importWallet('inst1', 'oid1', 'oid', 'oid5', 'password', '/tmp/ewallet.p12')
```

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 ''.

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**.



A Wallet is a Keystore that stores X.509 certificates and private keys in industry-standard, PKCS

self-signed certificate, click Create Self-Signed Wallet. To manage the contents of a wallet, selec

Deleting a wallet also deletes any certificate in the wallet. Any functionality that relies on the certificate will become unusable. Are you sure you want to delete the wallet selfsigned?

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', 'oid1', 'oid', 'selfsigned')
```

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:

- Log into Fusion Middleware Control using administrator credentials.
- Select the domain of interest.

> **Note:** You can use Setup to discover a specific Oracle WebLogic Server domain to work with.

- Use the navigation pane to locate the instance (for example, an Oracle HTTP Server instance) that will use the wallet.

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

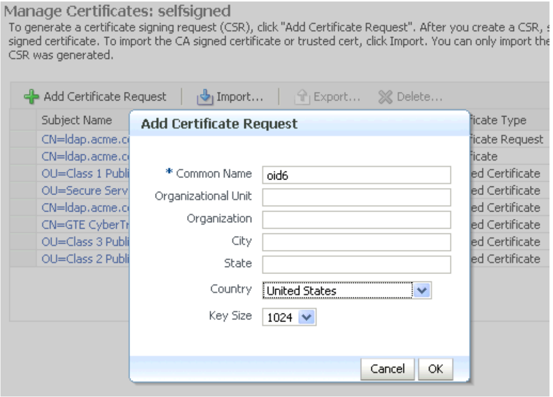
This section describes the following common certificate operations:

- 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**.
3. A dialog box appears where you enter the CRs DN values:



Manage Certificates: selfsigned

To generate a certificate signing request (CSR), click "Add Certificate Request". After you create a CSR, s
signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the
CSR was generated.

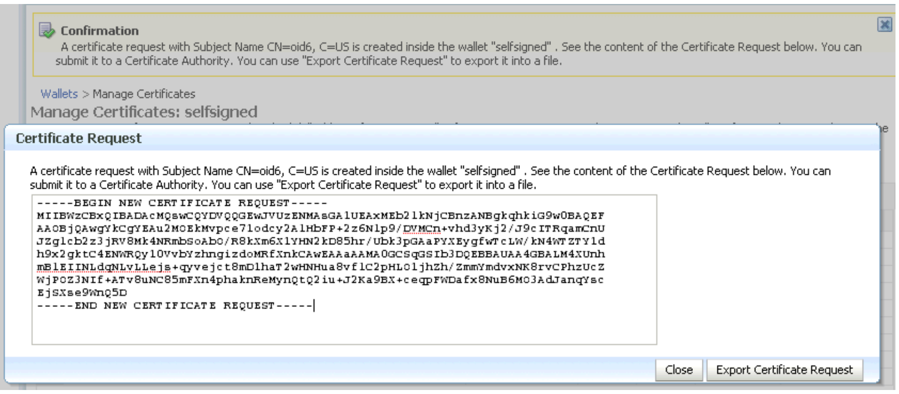
Subject Name	icate Type
CN=ldap.acme.c	icate Request
CN=ldap.acme.c	icate
OU=Class 1 Publi	ed Certificate
OU=Secure Serv	ed Certificate
CN=ldap.acme.c	ed Certificate
CN=GTE CyberTr	ed Certificate
OU=Class 3 Publi	ed Certificate
OU=Class 2 Publi	ed Certificate

Add Certificate Request

* Common Name	oid6
Organizational Unit	
Organization	
City	
State	
Country	United States
Key Size	1024

   Fields marked with an asterisk (*) are required.

4. Click **OK**.
5. 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.



Confirmation

A certificate request with Subject Name CN=oid6, C=US is created inside the wallet "selfsigned" . See the content of the Certificate Request below. You can submit it to a Certificate Authority. You can use "Export Certificate Request" to export it into a file.

Wallets > Manage Certificates

Manage Certificates: selfsigned

Certificate Request

A certificate request with Subject Name CN=oid6, C=US is created inside the wallet "selfsigned" . See the content of the Certificate Request below. You can submit it to a Certificate Authority. You can use "Export Certificate Request" to export it into a file.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBWzCBxQIBADAcMQswCQYDVQQGEwJVUzENMAsGA1UEAxMEb21kNjCBnzANBgkqhkiG9w0BAQEF
AAOBjQAwgYkCgYEAu2MOEkMvpce7lodcy2A1HbFP+2z6N1p9/DVMCn+vhd3yKj2/J9cITRqamCnU
JZglcb2z3jRV8Mk4NRmbSoAbO/R8kXm6X1YHN2kD85hr/Ubk3pGAaPYXEygfwTcLW/kN4WTZTY1d
h9x2gktC4ENWRQy10VvbYzhngizdoMRfXnkCAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBALM4XUnh
mBlEIINLdqNLvLLejs+qyvejct8mDlhaT2wHNHua8vflC2pHLO1jhZh/ZmmYmdvxNK8rvCPhzUcZ
WjPOZ3NIf+ATv8uNC85mFXn4phaknReMynQtQ2iu+J2Ka9BX+ceqpFWDafx8NuB6MO3AdJanqYsc
EjSXse9WnQ5D
-----END NEW CERTIFICATE REQUEST-----
```

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', 'oid1', 'oid', '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.

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', 'oid1', 'oid', '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`.

7.4.7.5 Importing a Certificate or a Trusted Certificate Using Fusion Middleware Control

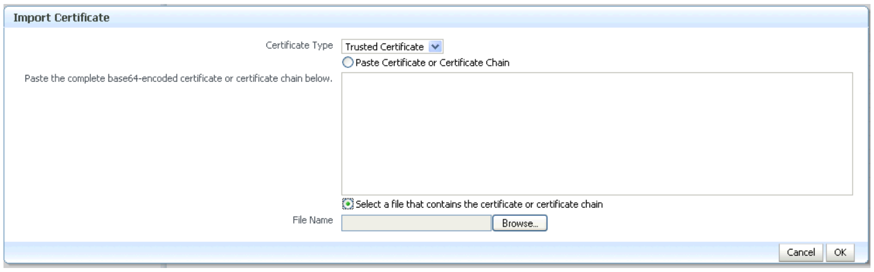
---

Note: You cannot use Fusion Middleware Control to import DER-encoded certificates or trusted certificates into an Oracle wallet. Use one of these tools instead:

- Oracle Wallet Manager or
- `orapki` command-line tool

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.



Import Certificate

Certificate Type: Trusted Certificate

○ Paste Certificate or Certificate Chain

Paste the complete base64-encoded certificate or certificate chain below.

● Select a file that contains the certificate or certificate chain

File Name

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 one of these tools instead:

- Oracle Wallet Manager or
- `orapki` command-line tool

Assuming the instance name is `inst1`, use this command to import a certificate into a wallet:

```
importWalletObject('inst1', 'oid1', 'oid', 'selfsigned', 'password',
'Certificate', '/tmp/cert.txt')
```

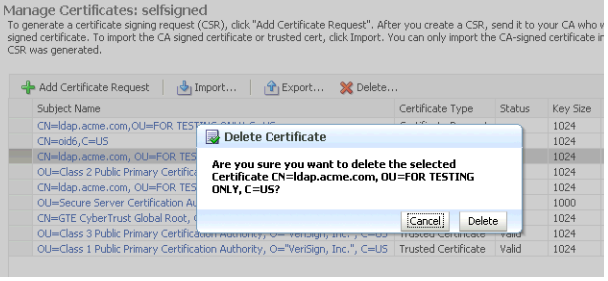
where `password` is the password for this wallet and `/tmp/cert.txt` is the file that contains BASE64 encoded certificate.

To import a trusted certificate, replace `Certificate` in the above command with `TrustedCertificate`.

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.



Manage Certificates: selfsigned

To generate a certificate signing request (CSR), click "Add Certificate Request". After you create a CSR, send it to your CA who w
signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CA-signed certificate ir
CSR was generated.

Subject Name	Certificate Type	Status	Key Size
CN=ldap.acme.com,OU=FOR TEST			1024
CN=oid6,C=US			1024
CN=ldap.acme.com, OU=FOR TES			1024
OU=Class 2 Public Primary Certifica			1024
CN=ldap.acme.com, OU=FOR TES			1024
OU=Secure Server Certification Au			1000
CN=GTE CyberTrust Global Root,			1024
OU=Class 3 Public Primary Certification Authority, O="VeriSign, Inc.", C=US	Trusted Certificate	Valid	1024
OU=Class 1 Public Primary Certification Authority, O="VeriSign, Inc.", C=US	Trusted Certificate	Valid	1024

Delete Certificate

Are you sure you want to delete the selected Certificate CN=ldap.acme.com, OU=FOR TESTING ONLY, C=US?

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', 'oid1', 'oid', '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`.

7.4.7.9 Converting a Self-Signed Certificate into a Third-Party Certificate Using Fusion Middleware Control

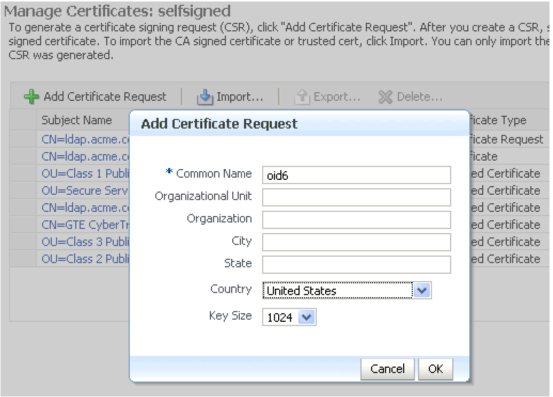
A self-signed certificate residing in a wallet can be converted into a third-party certificate signed by a certificate authority (CA). Take these steps to perform the task:

> **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 and Oracle Web Cache.

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:



Manage Certificates: selfsigned

To generate a certificate signing request (CSR), click "Add Certificate Request". After you create a CSR, s signed certificate. To import the CA signed certificate or trusted cert, click Import. You can only import the CSR was generated.

Subject Name	icate Type
CN=ldap.acme.c	icate Request
CN=ldap.acme.c	icate
OU=Class 1 Publi	ed Certificate
OU=Secure Serv	ed Certificate
CN=ldap.acme.c	ed Certificate
CN=GTE CyberTr	ed Certificate
OU=Class 3 Publi	ed Certificate
OU=Class 2 Publi	ed Certificate

Add Certificate Request

* Common Name	oid6
Organizational Unit	
Organization	
City	
State	
Country	United States
Key Size	1024

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:

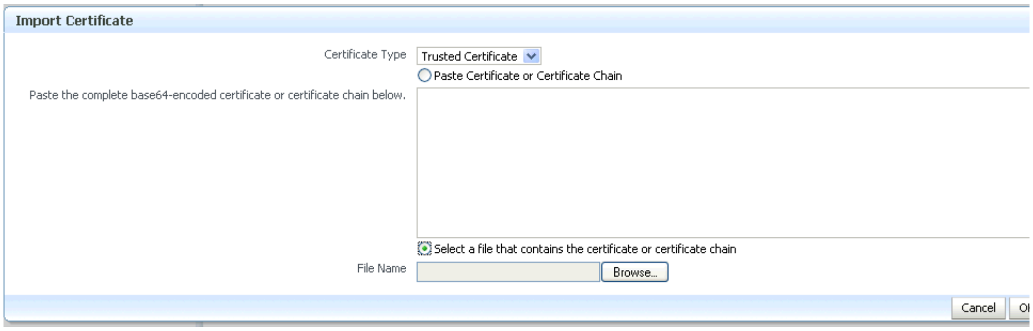
   - A single file containing both the newly generated certificate and its own CA certificate in `pkcs7` format
   - Two files, one containing the newly generated certificate and a second containing its own CA certificate

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 files:
     - Import the file containing the CA certificate as a trusted certificate (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.

   The CA returned a single file, which is imported as a trusted certificate:



Import Certificate

Certificate Type: Trusted Certificate

○ Paste Certificate or Certificate Chain

Paste the complete base64-encoded certificate or certificate chain below.

● Select a file that contains the certificate or certificate chain

File Name

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

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', 'oid1', 'oid', 'selfsigned', 'password',
   'subject_dn', 'key_size')
   ```

2. Export the certificate request:

   ```
   exportWalletObject('inst1', 'oid1', 'oid', '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 two separate files.

4. If you receive a single file from the CA, run the following command

   ```
   importWalletObject('inst1', 'oid1', 'oid', '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 files from the CA, import the CA certificate first as a trusted certificate, followed by the newly generated certificate.

   ```
   importWalletObject('inst1', 'oid1', 'oid', 'selfsigned', 'password',
   'TrustedCertificate', '/tmp/cacert.txt')

   importWalletObject('inst1', 'oid1', 'oid', '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.

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 for different components.

Root Directory for an Oracle Internet Directory Wallet

The root directory for wallets is `$ORACLE_INSTANCE/OID/admin`.

This root directory will contain subdirectories with wallet names; these subdirectories will contain the actual wallet files.

For example, assuming there are two wallets named `oid1` and `oid2`, respectively, a sample structure could look like:

```
$ORACLE_INSTANCE/OID/admin/oid1/cwallet.sso
$ORACLE_INSTANCE/OID/admin/oid1/ewallet.p12
$ORACLE_INSTANCE/OID/admin/oid2/cwallet.sso
```

Root Directory for an Oracle HTTP Server Wallet

The root directory for wallets is `$ORACLE_INSTANCE/config/OHS/`*`ohs_instance_name`*`/keystores`.

This root directory contains subdirectories with wallet names; these subdirectories contain the actual wallet files.

For example, assuming there are two wallets named `ohs1` and `ohs2`, respectively, a sample structure could look like:

```
$ORACLE_INSTANCE/config/OHS/ohs_instance1/keystores/ohs1/cwallet.sso
$ORACLE_INSTANCE/config/OHS/ohs_instance1/keystores/ohs1/ewallet.p12
$ORACLE_INSTANCE/config/OHS/ohs_instance1/keystores/ohs2/cwallet.sso
```

Root Directory for an Oracle Web Cache Wallet

The root directory for wallets is `$ORACLE_INSTANCE/config/WebCache/`*`webcache_instance_name`*`/keystores`.

This root directory will contain subdirectories with wallet names; these subdirectories will contain the actual wallet files.

For example, assuming there are two wallets named `wc1` and `wc2`, respectively, a sample structure could look like:

```
$ORACLE_INSTANCE/config/WebCache/webcache_instance1/keystores/wc1/cwallet.sso
$ORACLE_INSTANCE/config/WebCache/webcache_instance1/keystores/wc1/ewallet.p12
$ORACLE_INSTANCE/config/WebCache/webcache_instance1/keystores/wc2/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.

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).

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.

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.

Part IV

Deploying Applications

This part describes the deployment process and how to deploy applications to Oracle Fusion Middleware.

Part IVcontains the following chapters:

- Chapter 8, "Understanding the Deployment Process"
- Chapter 9, "Deploying Applications"

8 Understanding the Deployment Process

This chapter describes the following topics:

- What Is a Deployer?
- General Procedures for Moving from Application Design to Production Deployment
- Diagnosing Typical Problems

8.1 What Is a Deployer?

A **deployer** is responsible for deploying applications, such as Java EE applications, ADF applications, SOA Composite applications, or WebCenter 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:

- *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*
- *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
- *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

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 *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server*
- For Oracle ADF, see "*Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
- For Oracle WebCenter, see *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*
- For Oracle SOA Suite, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

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, ADF, and WebCenter applications, you package the application in an EAR file. For Oracle SOA Suite, you package the application into a JAR or ZIP file.

   For information about packaging the application, see:

   - For Java EE applications: "Packaging Files for Deployment" in the *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server*
   - For Oracle ADF: "What You May Need to Know About EAR Files and Packaging" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
   - For Oracle WebCenter: "Packaging and Deploying a Custom WebCenter Application to a WebLogic Managed Server" in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*
   - For Oracle SOA Suite: "Understanding the Packaging Impact" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

2. Set up your environment. This includes:

   - Installing and configuring a domain and a Managed Server that is configured with the correct domain template. For example, if you are deploying an Oracle

SOA Suite application, the Managed Server must use the Oracle SOA Suite domain template. The appropriate domain template is applied when you create the domain using the Configuration Wizard. Alternatively, you can extend a domain to use another domain template, as described in Section 18.2.

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 the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
- For Oracle WebCenter: *Oracle Fusion Middleware Installation Guide for Oracle WebCenter* and "Preparing the Target Environment" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*
- For Oracle SOA Suite: *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*

- Creating any necessary schemas in an existing database. See *Oracle Fusion Middleware Repository Creation Utility User's Guide*.
- Registering the MDS Repository with the Oracle WebLogic Server domain, if your application uses the MDS Repository. For example, Oracle SOA Suite and Oracle WebCenter applications require MDS. Some ADF applications involve customizations using MDS. See Section 13.3.2.1 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: *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*
   - For Oracle ADF: "How to Create a JDBC Data Source for Oracle WebLogic Server" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
   - For Oracle WebCenter: "Configuring Data Sources" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*
   - For Oracle SOA Suite: "Creating Data Sources and Queues" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

4. For Oracle SOA Suite, create connection factories and connection pooling. For more information, see "Creating Connection Factories and Connection Pooling" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

5. 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 the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
   - For Oracle WebCenter: "Configuring Connections" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*

- For Oracle SOA Suite: "Creating an Application Server Connection" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

6. For Oracle SOA Suite, create a SOA-MDS connection, if the SOA composite application shares metadata with other composites. See "Creating a SOA-MDS Connection" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

7. 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 the *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server*
   - For Oracle SOA Suite: "Introduction to Configuration Plans" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

8. Migrate application security, such as credentials, identities, and policies. For more information, see:

   - For pure Java EE applications: "Migrating Security Data" in the *Oracle Fusion Middleware Securing Oracle WebLogic Server*
   - For Oracle ADF: "Preparing the Secure Application for Deployment" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
   - For Oracle WebCenter: "Configuring Security" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*
   - For Oracle SOA Suite: "Enabling Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

9. Create a deployment profile. A **deployment profile** packages or archives a custom ADF, WebCenter, or SOA application and associated files so that the application can be deployed to an Oracle WebLogic Server Managed Server instance. Deployment profiles are created at project and application level.

   For more information, see:

   - For Oracle ADF: "How to Create Deployment Profiles" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
   - For Oracle WebCenter: "Creating Deployment Profiles" in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*
   - For Oracle SOA Suite: "Optionally Creating a Project Deployment Profile" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

10. Migrate Oracle JDeveloper extensions for Oracle SOA Suite and Oracle WebCenter. Table 8–1 shows the extensions and where they are documented:

Table 8–1 Oracle JDeveloper Extensions

Component	Extension	See
Oracle WebCenter	WebCenter extensions	"Creating and Provisioning a WebLogic Managed Server Instance" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*

Table 8–1 (Cont.) Oracle JDeveloper Extensions

Component	Extension	See
Oracle SOA Suite	SOA extensions	"Installing Additional Oracle Fusion Middleware Design Time Components" in the *Oracle Fusion Middleware Installation Guide for Oracle JDeveloper*

11. 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 the *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server*
    - For Oracle ADF: "Deploying the Application" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
    - For Oracle WebCenter: "Deploying the Application to a WebLogic Managed Server Instance" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*
    - For Oracle SOA Suite: "Deploying SOA Composite Applications" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

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 the *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
- For Oracle ADF: "Deploying Using Scripts and Ant" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Application Development Framework*
- For Oracle WebCenter: "Creating and Provisioning a WebLogic Managed Server Instance" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*
- For Oracle SOA Suite:"Managing SOA Composite Applications with Scripts" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

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 Steps 4, 5, and 6 in Section 8.2.2.
- Oracle JDeveloper extensions. Ensure that you have migrated any Oracle JDeveloper extensions. See Table 8–1.
- 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 migrates application security, such as credentials, identities, and policies. See Step 8 in Section 8.2.2.

In addition, see the following for information about troubleshooting SOA applications:

- "Troubleshooting Common Deployment Errors" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

9 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, such as Java EE or SOA applications, to Oracle Fusion Middleware.

It contains the following topics:

- Overview of Deploying Applications
- Understanding Data Sources
- Deploying, Undeploying, and Redeploying Java EE Applications
- Deploying, Undeploying, and Redeploying Oracle ADF Applications
- Deploying, Undeploying, and Redeploying SOA Composite Applications
- Deploying, Undeploying, and Redeploying WebCenter Applications
- 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:

- Understanding Deployment, Redeployment, and Undeployment
- What Types of Applications Can You Deploy?

9.1.1 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. 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 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.

9.1.2 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.
- An Oracle SOA Suite composite application. A SOA composite application is a single unit of deployment that greatly simplifies the management and lifecycle of SOA applications.
- An Oracle WebCenter application. WebCenter applications differ from traditional Java EE applications in that they support run-time customization, including the application's pages, the portlets contained within these pages, and document libraries.

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. The following application types use a MAR as a container for content that is deployed to the MDS Repository: ADF applications, SOA composite applications, and Oracle WebCenter applications.

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

Table 9–1 (Cont.) Tools to Deploy Applications

Type of Application	Tools to Use
SOA Composite application	Fusion Middleware Control: SOA Composite Deployment Wizard Oracle JDeveloper WLST command line
WebCenter 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 B2B, and 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 *Oracle Fusion Middleware Security Guide*.

9.2 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 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 Also:** *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server* for more information about data sources

When you configure certain Oracle Fusion Middleware components, such as Oracle SOA Suite or Oracle WebCenter, 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 Also:** *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard* for information about specifying data sources with the Configuration Wizard

To create an MDS data source, 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.

If you are using RAC or Oracle Fusion Middleware Cold Failover Cluster, you must configure multi data sources. To do so, 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.

> **See Also:** *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server* for more information about configuring multi data sources

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 Also:** *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server* for information about deploying using Oracle WebLogic Server Administration Console and for more information about using 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.

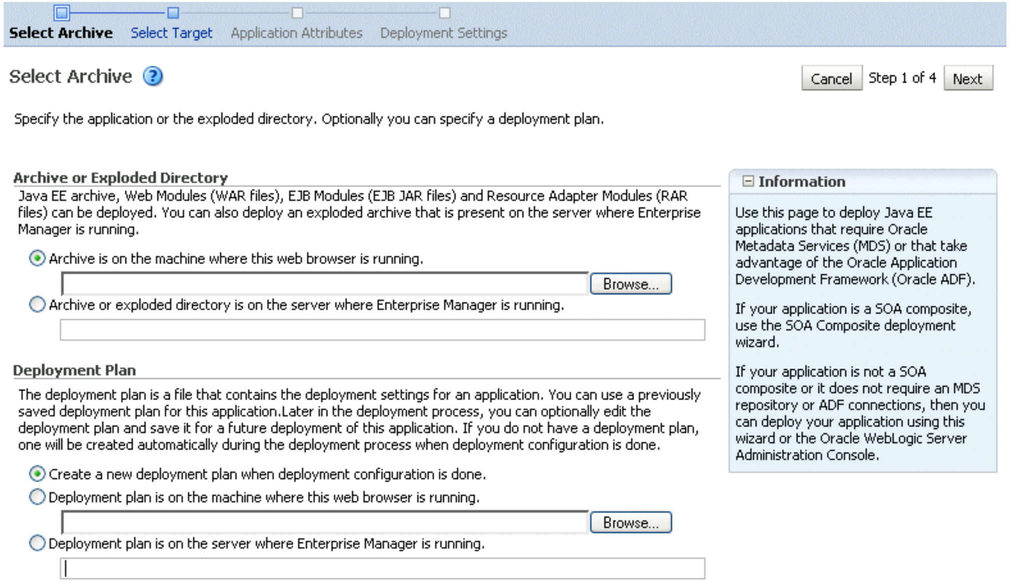
9.3.1.1 Deploying Java EE Applications Using Fusion Middleware Control

To deploy a Java EE application to a Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **WebLogic Domain,** and then 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 **Application Deployment,** then **Deploy.**

   The Deployment Wizard, Select Archive page is displayed, as shown in the following figure:



Select Archive

Specify the application or the exploded directory. Optionally you can specify a deployment plan.

Archive or Exploded Directory

Java EE archive, Web Modules (WAR files), EJB Modules (EJB JAR files) and Resource Adapter Modules (RAR 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.

● 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.

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 deployment wizard.

If your application is not a SOA composite or it does not require an MDS repository or ADF connections, then you can deploy your application using this wizard or the Oracle WebLogic Server Administration Console.

4. 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.
5. 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.
6. Click **Next.**

   The Select Target page is displayed.
7. 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.

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. In the Distribution section, you can select one of the following:

    - **Distribute and start application (servicing all requests)**
    - **Distribute and start application in admin mode (servicing only admin requests)**
    - **Distribute only**

12. You can expand Other Options, which provides the following options:

    - Use the defaults defined by the deployment's targets. Recommended selection.
    - Copy this application onto every target. During deployment, the files are copied automatically to the Managed Servers to which the application is targeted.

13. Click **Next.**

    The Deployment Wizard, 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. Depending on the type of application, 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 application security. Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed.

      If the application contains jazn-data.xml or cwallet.sso, the Configure Application Security page displays the following sections:

      – If it contains jazn-data.xml, the page displays the Application Policy Migration section.

        The Application Policy Migration section contains the following options:

        To append, overwrite, or ignore policy migration. Select **Append** when deploying the application for the first time. If the application was previously deployed and the application authorization policy exists, select either **Append**, or **Ignore** to keep the application authorization policy. Select **Overwrite** to overwrite the previous policy.

To specify the Application stripe ID, if the stripe ID is inconsistent with the one defined in the migration options.

To remove the policies when the application is undeployed.

- If it contains cwallet.sso, the page displays the Application Credential Migration section.
- If it contains both, the page displays both sections.

For information about these settings, see "Deploying JavaEE and Oracle ADF Applications with Fusion Middleware Control" in the *Oracle Fusion Middleware Security Guide*.

If the application contains neither of these files, 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 EJB modules: Click **Go to Task** in the Configure EJB modules row. 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.

15. 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.

16. Click **Deploy.**

    Fusion Middleware Control displays processing messages.

17. When the deployment is completed, click **Close.**

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_
10.3/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 weblogic -password weblogic -deploy
   -name myApp c:\localfiles\mainWebApp
   -plan c:\localfiles\productionEnvPlan.xml
```

See Also:

- "Deployment Tools" in *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server* for more information about using WLST to deploy applications
- The *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

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.

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 Application Deployment menu, choose **Application Deployment,** then **Undeploy.**

   The confirmation page is displayed.
4. Click **Undeploy.**

   Processing messages are displayed.
5. 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. This section describes how to redeploy an application to a Managed Server.

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 the farm, then **Application Deployments.**
2. Select the application.

   The application home page is displayed.
3. From the Application Deployment menu, choose **Application Deployment,** and then **Redeploy.**

   The Select Application page is displayed.
4. Click **Next.**
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. Click **Next.**

   The Application Attributes page is displayed.
8. Click **Next.**

   The Deployment Wizard, Deployment Settings page is displayed.
9. 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, you can:

- Configure Web modules
- Configure application security
- Configure EJB modules

10. 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.

11. Click **Redeploy.**

    Processing messages are displayed.

12. When the operation completes, click **Close.**

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

The 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.

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 Also: *Oracle Fusion Middleware Fusion Developer's Guide for 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 farm, then **WebLogic Domain,** and then the domain.
2. Select the server in which you want to deploy the application.

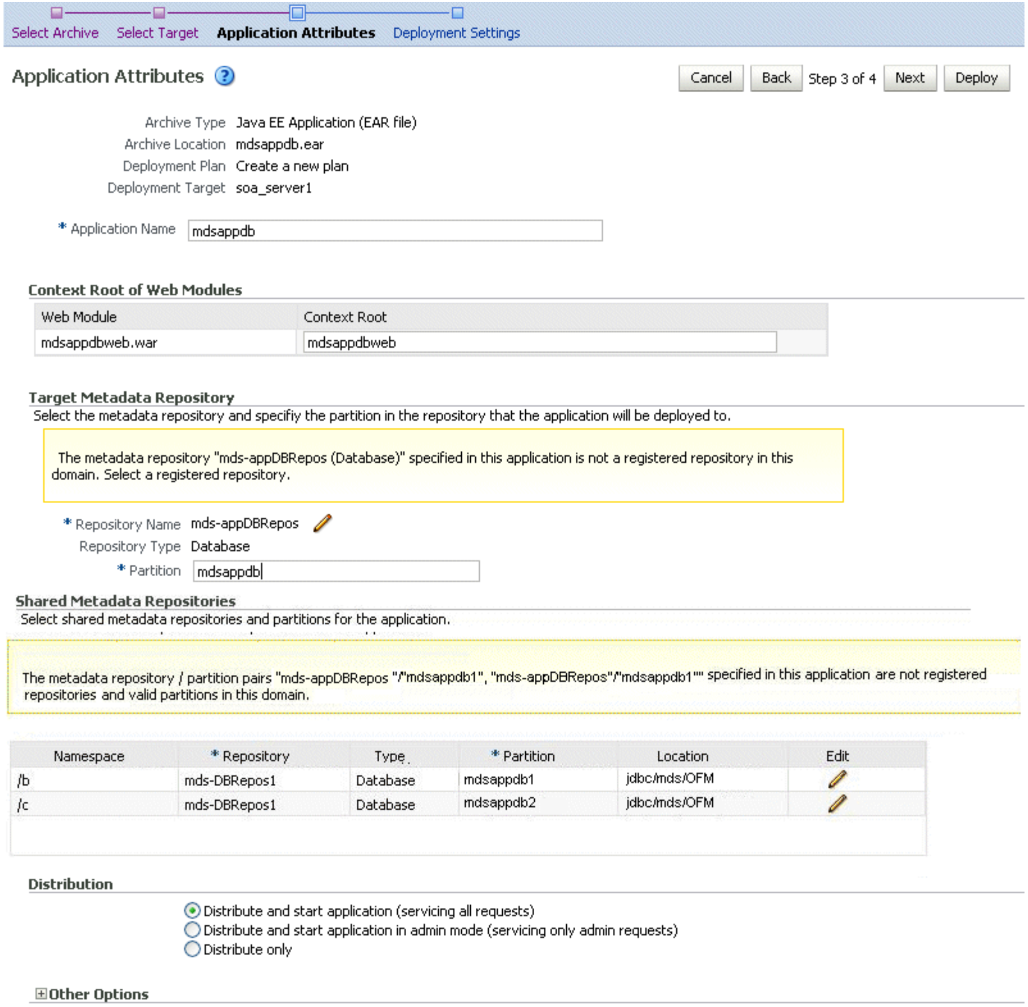
   The server home page is displayed.
3. From the WebLogic Server menu, choose **Application Deployment,** then **Deploy.**

   The Deployment Wizard, Select Archive page is displayed.
4. 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.
5. 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.
6. Click **Next.**

   The Select Target page is displayed.
7. 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.
8. Click **Next.**

   The Application Attributes page is displayed, as shown in the following figure:



Application Attributes

Archive Type	Java EE Application (EAR file)
Archive Location	mdsappdb.ear
Deployment Plan	Create a new plan
Deployment Target	soa_server1
* Application Name	mdsappdb

Context Root of Web Modules

Web Module	Context Root
mdsappdbweb.war	mdsappdbweb

Target Metadata Repository

Select the metadata repository and specifiy the partition in the repository that the application will be deployed to.

The metadata repository "mds-appDBRepos (Database)" specified in this application is not a registered repository in this domain. Select a registered repository.

* Repository Name	mds-appDBRepos
Repository Type	Database
* Partition	mdsappdb

Shared Metadata Repositories

Select shared metadata repositories and partitions for the application.

The metadata repository / partition pairs "mds-appDBRepos "/"mdsappdb1", "mds-appDBRepos"/"mdsappdb1"" specified in this application are not registered repositories and valid partitions in this domain.

Namespace	* Repository	Type	* Partition	Location	Edit
/b	mds-DBRepos1	Database	mdsappdb1	jdbc/mds/OFM	
/c	mds-DBRepos1	Database	mdsappdb2	jdbc/mds/OFM	

Distribution

● Distribute and start application (servicing all requests)
○ Distribute and start application in admin mode (servicing only admin requests)
○ Distribute only

⊞ Other Options

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. 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.

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. 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.

13. In the Distribution section, you can select one of the following:

    - **Distribute and start application (servicing all requests)**
    - **Distribute and start application in admin mode (servicing only admin requests)**
    - **Distribute only**

14. Click **Next.**

    The Deployment Wizard, Deployment Settings page is displayed.

15. 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, 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 application security. Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed.

      If the application contains jazn-data.xml or cwallet.sso, the Configure Application Security page displays the following sections:

      – If it contains jazn-data.xml, the page displays the Application Policy Migration section.

      – If it contains cwallet.sso, the page displays the Application Credential Migration section.

      – If it contains both, the page displays both sections.

      For information about the settings in this page, see "Deploying JavaEE and Oracle ADF Applications with Fusion Middleware Control" in the *Oracle Fusion Middleware Security Guide.*

      If the application contains neither of these files, 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 EJB modules: Click **Go to Task** in the Configure EJB modules row. 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 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 *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

16. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.

17. Click **Deploy.**

    Fusion Middleware Control displays processing messages.

18. 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 example 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 *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server* for more information about using WLST to deploy applications
- The *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

To deploy the application using the Oracle WebLogic Server Administration Console:

a. If you have not already done so, in the Change Center of the Administration Console, click **Lock & Edit**.

b. In the left pane of the Administration Console, select **Deployments.**

c. In the right pane, click **Install.**

See Also: The Help in the Oracle WebLogic Server Administration Console.

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,** then the application to undeploy.

   The application home page is displayed.

2. From the Application Deployment menu, choose **Application Deployment,** then **Undeploy.**

   The confirmation page is displayed.

3. Click **Undeploy.**

   Processing messages are displayed.

4. 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.

To redeploy an Oracle ADF application using Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **Application Deployments.**
2. Select the application.

   The application home page is displayed.
3. From the Application Deployment menu, choose **Application Deployment,** and then **Redeploy.**

   The Select Application page is displayed.
4. Click **Next.**

   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. Click **Next.**

   The Application Attributes page is displayed.
8. In the Application Attributes section, for **Application Name,** enter the application name.
9. 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.
10. 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.
11. 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.
12. Click **Next.**

    The Deployment Settings page is displayed.

13. 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. You can:
    - Configure Web modules
    - Configure application security
14. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.
15. Click **Deploy.**

    Fusion Middleware Control displays processing messages.
16. When the deployment is completed, click **Close.**
17. In the Confirmation page, click **Redeploy.**

9.5 Deploying, Undeploying, and Redeploying SOA Composite Applications

SOA composite applications consist of the following:

- Service components such as Oracle Mediator for routing, BPEL processes for orchestration, human tasks for workflow approvals, business rules for designing business decisions, and complex event processing (CEP) for queries of event streams
- Binding components (services and references) for connecting SOA composite applications to external services, applications, and technologies

These components are assembled together into a SOA composite application. This application is a single unit of deployment that greatly simplifies the management and lifecycle of SOA applications.

You can use Fusion Middleware Control, Oracle JDeveloper, or the command line to deploy, undeploy, or redeploy a SOA application. The following topics describe using Fusion Middleware Control to accomplish these tasks:

- Deploying SOA Composite Applications
- Undeploying SOA Composite Applications
- Redeploying SOA Composite Applications

> **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite*

9.5.1 Deploying SOA Composite Applications

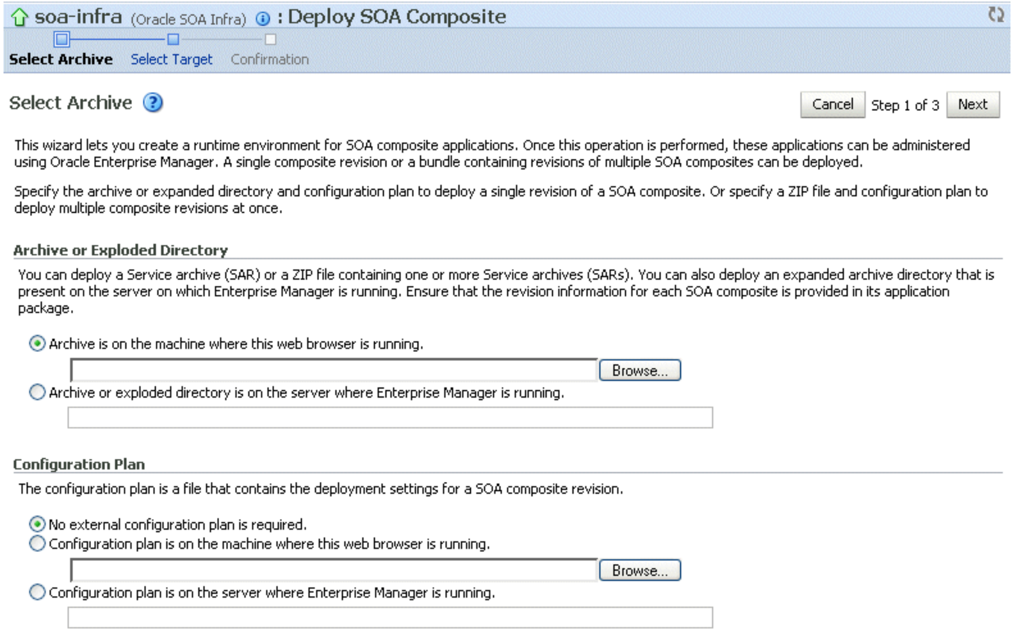
When you deploy a SOA composite application, the deployment extracts and activates the composite application in the SOA Infrastructure.

You can deploy SOA composite applications from Fusion Middleware Control with the Deploy SOA Composite wizard:

1. From the navigation pane, expand the farm, then **SOA,** and then select **soa-infra**.

   The SOA Infrastructure home page is displayed.
2. From the SOA Infrastructure menu, choose **SOA Deployment,** then **Deploy.**

   The Deployment Wizard, Select Archive page is displayed, as shown in the following figure:



Select Archive

This wizard lets you create a runtime environment for SOA composite applications. Once this operation is performed, these applications can be administered using Oracle Enterprise Manager. A single composite revision or a bundle containing revisions of multiple SOA composites can be deployed.

Specify the archive or expanded directory and configuration plan to deploy a single revision of a SOA composite. Or specify a ZIP file and configuration plan to deploy multiple composite revisions at once.

Archive or Exploded Directory

You can deploy a Service archive (SAR) or a ZIP file containing one or more Service archives (SARs). You can also deploy an expanded archive directory that is present on the server on which Enterprise Manager is running. Ensure that the revision information for each SOA composite is provided in its application package.

● Archive is on the machine where this web browser is running.

Browse...

○ Archive or exploded directory is on the server where Enterprise Manager is running.

Configuration Plan

The configuration plan is a file that contains the deployment settings for a SOA composite revision.

● No external configuration plan is required.

○ Configuration plan is on the machine where this web browser is running.

Browse...

○ Configuration plan is on the server where Enterprise Manager is running.

3. In the Archive or Exploded Directory section, specify the archive of the SOA composite application to deploy. The archive contains the project files of the application to be deployed (for example, **HelloWorld_rev1.0.jar** for a single archive or **OrderBooking_rev1.0.zip** for multiple archives).
4. In the Configuration Plan section, optionally specify the configuration plan to include with the archive. The configuration plan enables you to define the URL and property values to use in different environments. During the process deployment, the configuration plan is used to search the SOA project for values that must be replaced to adapt the project to the next target environment.
5. Click **Next**.

   The Select Target page appears.
6. Select the Managed Server or cluster to which to deploy the SOA composite application archive. You can deploy to multiple servers and clusters.
7. Click **Next**.

   The Confirmation page appears.
8. Review your selections.
9. Select whether or not to deploy the SOA composite application as the default revision. The default revision is instantiated when a new request comes in.
10. Click **Deploy**.

    Processing messages are displayed.
11. When deployment has completed, click **Close**.

> **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite* for complete information about deploying SOA Composite applications

9.5.2 Undeploying SOA Composite Applications

You can undeploy SOA composite applications from Fusion Middleware Control with the Undeploy SOA Composite wizard:

1. From the navigation pane, expand the farm, then **SOA,** and then select **soa-infra.**

   The SOA Infrastructure home page is displayed.

2. From the SOA Infrastructure menu, choose **SOA Deployment,** then **Undeploy.**

   The Confirmation page is displayed.

3. Review your selections.
4. If you are satisfied, click **Undeploy**.

   Processing messages are displayed.

5. When undeployment has completed, click **Close**.

> **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite* for complete information about deploying SOA Composite applications

9.5.3 Redeploying SOA Composite Applications

You can redeploy SOA composite applications from Fusion Middleware Control with the Redeploy SOA Composite wizard:

1. From the navigation pane, expand the farm, then **SOA,** and then select **soa-infra.**

   The SOA Infrastructure home page is displayed.

2. From the SOA Infrastructure menu, choose **SOA Deployment,** then **Redeploy.**

   The Select Composite page is displayed.

3. Select the composite that you want to redeploy.
4. In the Archive or Exploded Directory section, select the location of the SOA composite application revision you want to redeploy.
5. Click **Next.**

   The Select Archive page appears.

6. In the Configuration Plan section, optionally specify the configuration plan to include with the archive.
7. Click **Next**.

   The Confirmation page appears.

8. Select whether or not to redeploy the SOA composite application as the default revision.
9. Click **Redeploy**.

   Processing messages are displayed.

10. When redeployment has completed, click **Close**.

> **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite* for complete information about deploying SOA Composite applications

9.6 Deploying, Undeploying, and Redeploying WebCenter Applications

WebCenter applications differ from traditional Java EE applications in that they support run-time customization, such as the application's pages, the portlets contained within these pages, and the document libraries. Customizations are stored as follows:

- WebCenter application customizations are stored in Oracle Metadata Service (MDS), which is installed in a database.
- Portlet producer customizations (or preferences) are usually stored in a database preference store.

You can use Fusion Middleware Control, Oracle JDeveloper, or the command line to deploy, undeploy, or redeploy a WebCenter application. The following topics describe using Fusion Middleware Control to accomplish these tasks:

- Deploying WebCenter Applications
- Undeploying WebCenter Applications
- Redeploying WebCenter Applications

> **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*

9.6.1 Deploying WebCenter Applications

To deploy your application to a Managed Server that resides outside JDeveloper, you must first create application deployment plans. In Oracle JDeveloper, first create a project-level deployment profile and then an application-level deployment profile. The project-level deployment profile is packaged as a Web Application Archive (WAR) file. The application-level deployment profile is packaged as a Metadata Archive (MAR). A MAR is a compressed archive of selected metadata. A single MAR can contain metadata content of multiple projects. MAR files are used to deploy metadata content to the MDS Repository. For information about creating deployment plans with Oracle JDeveloper, see *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

To deploy an Oracle WebCenter application to a Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **WebLogic Domain,** and then 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, select **Application Deployment,** then **Deploy.**

   The Deployment Wizard, Select Archive page is displayed.
4. 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.
5. 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.

6. Click **Next.**

   The Select Target page is displayed.

7. 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.

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, specify the context root for your application if you have not specified it in application.xml. 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. In the Target Metadata Repository 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.

    - 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.

      Each application must have a unique partition in the repository.

12. Click **Next.**

    The Deployment Wizard, Deployment Settings page is displayed.

13. 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, 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 application security. Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed.

      If the application contains jazn-data.xml or cwallet.sso, the Configure Application Security page displays the following sections:

      – If it contains jazn-data.xml, the page displays the Application Policy Migration section.

- If it contains cwallet.sso, the page displays the Application Credential Migration section.
- If it contains both, the page displays both sections.

For information about the settings in this page, see "Deploying JavaEE and Oracle ADF Applications with Fusion Middleware Control" in the *Oracle Fusion Middleware Security Guide*.

If the application contains neither of these files, 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 EJB modules: Click **Go to Task** in the Configure EJB modules row. 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 references.

  For example, you can configure the maximum number of beans in the free pool or the network access point.

- 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 more information about setting these attributes, see the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*.

14. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.

15. Click **Deploy.**

    Fusion Middleware Control displays processing messages.

16. When the deployment is completed, click **Close.**

9.6.2 Undeploying WebCenter Applications

To undeploy a WebCenter Application:

1. From the navigation pane, expand **Application Deployments,** then the application to undeploy.

   The application home page is displayed.

2. From the Application Deployment menu, select **Application Deployment,** then **Undeploy.**

   The confirmation page is displayed.

3. Click **Undeploy.**

Processing messages are displayed.

4. 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.

9.6.3 Redeploying WebCenter Applications

To redeploy a WebCenter Application:

1. From the navigation pane, expand the farm, then **WebLogic Domain,** and then 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, select **Application Deployment,** then **Redeploy.**

   The Select Application page is displayed. You can only redeploy applications that are versioned. If the application is not versioned, you must undeploy, then redeploy.
4. Select the application to 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, specify the context root for your application if you have not specified it in application.xml. 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. In the Target Metadata Repository section, select the MDS Repository and for **Partition Name,** enter a partition name. If the partition or repository specified in the application is not valid in the domain, Fusion Middleware Control displays a message.

12. Click **Next.**

    The Deployment Settings page is displayed.

13. 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. You can:

    - Configure Web modules
    - Configure application security

14. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.

15. Click **Redeploy.**

    Fusion Middleware Control displays processing messages.

16. When the deployment is completed, click **Close.**

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 this configuration persist in MDS as customizations. Because these persist as customizations:
>
> - Any changes made to this configuration are retained across application deployments. For example, if a configuration attribute is changed and if the application is redeployed to the same partition, the previously changed attribute value is still retained.
> - In a cluster, if an attribute is changed on one Managed Server, that change is automatically propagated to all Managed Servers in the cluster.

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 the farm, then **Application Deployments.** Then, select an application.

   The application's home page is displayed.

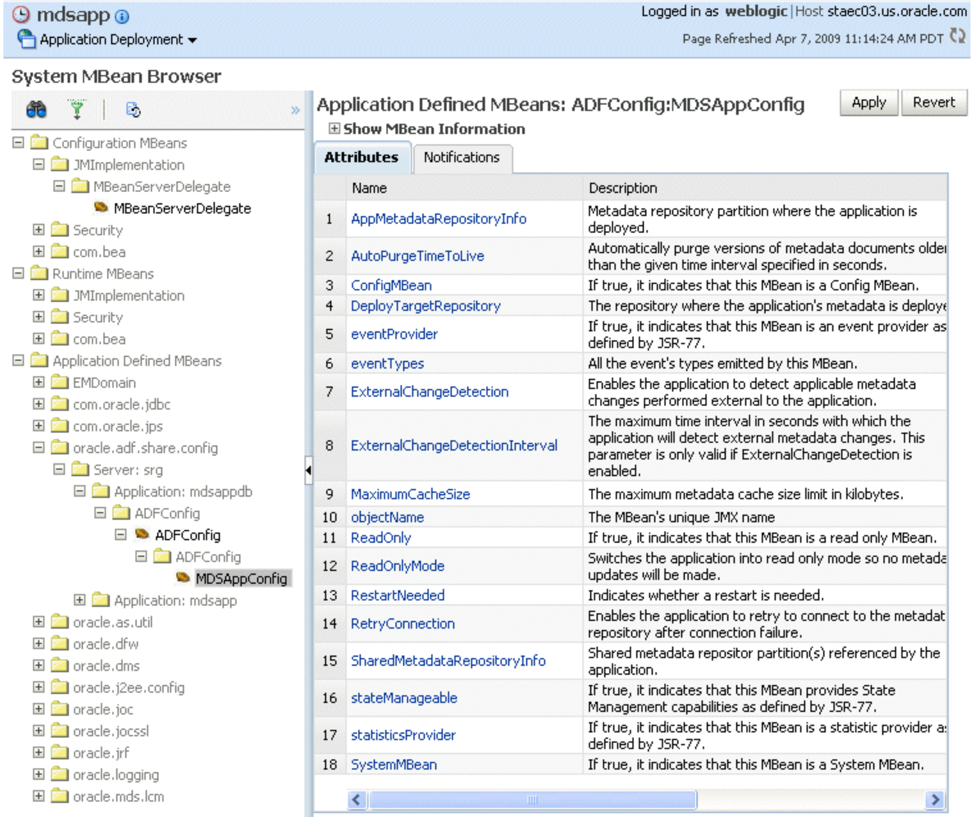
2. From the Application Deployment menu, choose **System MBean Browser.**

   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 application is deployed.
2	AutoPurgeTimeToLive	Automatically purge versions of metadata documents older than the given time interval specified in seconds.
3	ConfigMBean	If true, it indicates that this MBean is a Config MBean.
4	DeployTargetRepository	The repository where the application's metadata is deploye
5	eventProvider	If true, it indicates that this MBean is an event provider as defined by JSR-77.
6	eventTypes	All the event's types emitted by this MBean.
7	ExternalChangeDetection	Enables the application to detect applicable metadata changes performed external to the application.
8	ExternalChangeDetectionInterval	The maximum time interval in seconds with which the application will detect external metadata changes. This parameter is only valid if ExternalChangeDetection is enabled.
9	MaximumCacheSize	The maximum metadata cache size limit in kilobytes.
10	objectName	The MBean's unique JMX name
11	ReadOnly	If true, it indicates that this MBean is a read only MBean.
12	ReadOnlyMode	Switches the application into read only mode so no metada updates will be made.
13	RestartNeeded	Indicates whether a restart is needed.
14	RetryConnection	Enables the application to retry to connect to the metadat repository after connection failure.
15	SharedMetadataRepositoryInfo	Shared metadata repositor partition(s) referenced by the application.
16	stateManageable	If true, it indicates that this MBean provides State Management capabilities as defined by JSR-77.
17	statisticsProvider	If true, it indicates that this MBean is a statistic provider a defined by JSR-77.
18	SystemMBean	If true, it indicates that this MBean is a System MBean.

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.
DeployTargetRepository	The name of the target repository configured for the application.

Table 9–2 (Cont.) MDS Configuration Attributes for Deployed Applications

Attribute	Description
ExternalChangeDetection	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. 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. 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. 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. This attribute is applicable only to database-based repositories. The default is true.
ExternalChangeDetectionInterval	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. 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. The default is 30 seconds.
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.
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.
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.

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().getObjectName()
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().getObjectName()
mbs.invoke(adfConfigMBean, 'save', None, None)
```

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:

10

Monitoring Oracle Fusion Middleware

This chapter describes how to monitor Oracle Fusion Middleware using Oracle Enterprise Manager Fusion Middleware Control, Oracle WebLogic Server Administration Console, and the command line. It describes the following topics:

- 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:

- You can monitor the status of Oracle WebLogic Server domains, servers, Java components, and applications using Oracle WebLogic Server Administration Console. From the Administration Console, navigate to the entity's page. See "Overview of the Administration Console" in the *Oracle Fusion Middleware Introduction to Oracle WebLogic Server* for information on monitoring using the console.
- You can monitor the status of Oracle WebLogic Server domains, servers, Java components, system components, and applications using Fusion Middleware Control. Navigate to the entity's home page, for example, to the home page for an Oracle HTTP Server instance.
- You can monitor the status of your environment using the command line.

  To monitor the status of Java components with the command line, use the WLST `state` command, using 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 with the command line, use the `opmnctl status` command, using the following format:

  ```
  opmnctl status  [scope] [options]
  ```

For example, to view the status of all processes monitored by OPMN, use the following command:

```
opmnctl status
```

The following topics provide more detail:

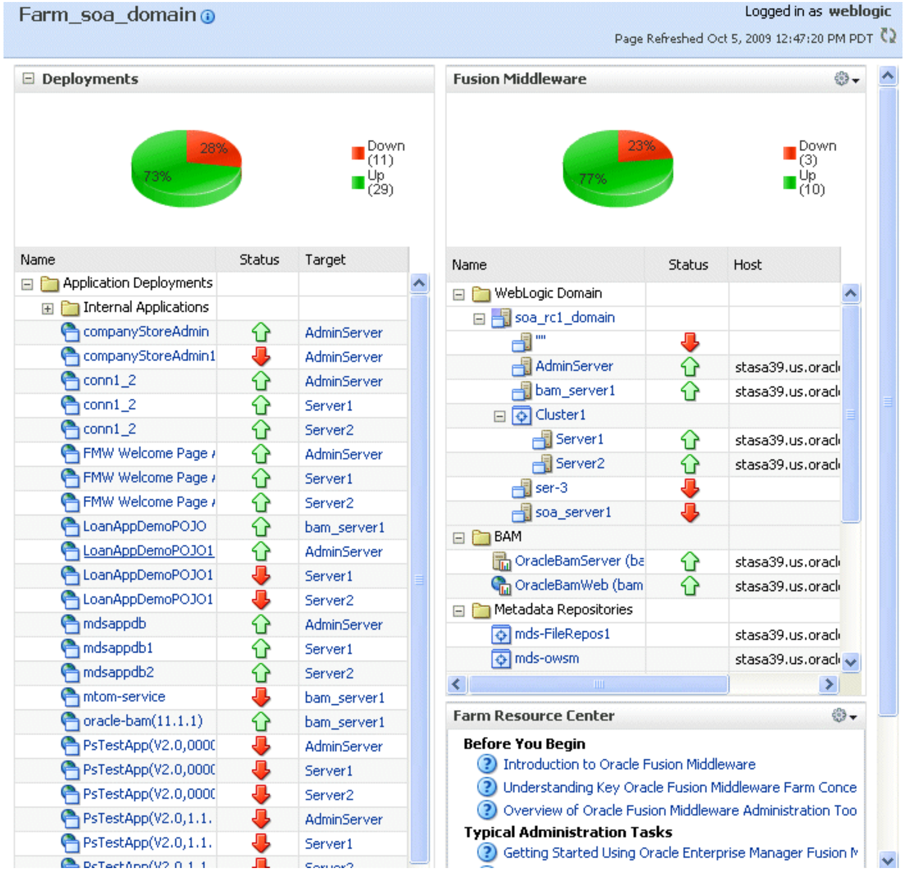
- Viewing General Information
- Monitoring an Oracle WebLogic Server Domain
- Monitoring an Oracle WebLogic Server Administration or Managed Server
- Monitoring a Cluster
- Monitoring a Component
- Monitoring Java EE Applications
- Monitoring ADF Applications
- Monitoring SOA Composite Applications
- Monitoring Oracle WebCenter Applications

10.1.1 Viewing General Information

You can view the overall status of the Oracle Fusion Middleware environment from the home page of the farm using Fusion Middleware Control. This page lists the availability of all components, an application deployment summary, including SOA composites, if any SOA composite applications are deployed.

To view the status, from the navigation pane, select the farm.

The farm home page is displayed, as shown in the following figure:



Farm_soa_domain

Logged in as **weblogic**

Page Refreshed Oct 5, 2009 12:47:20 PM PDT

Deployments

Name	Status	Target
Application Deployments		
Internal Applications		
companyStoreAdmin	↑	AdminServer
companyStoreAdmin1	↓	AdminServer
conn1_2	↑	AdminServer
conn1_2	↑	Server1
conn1_2	↑	Server2
FMW Welcome Page /	↑	AdminServer
FMW Welcome Page /	↑	Server1
FMW Welcome Page /	↑	Server2
LoanAppDemoPOJO	↑	bam_server1
LoanAppDemoPOJO1	↑	AdminServer
LoanAppDemoPOJO1	↓	Server1
LoanAppDemoPOJO1	↓	Server2
mdsappdb	↑	AdminServer
mdsappdb1	↑	Server1
mdsappdb2	↑	Server2
mtom-service	↓	bam_server1
oracle-bam(11.1.1)	↑	bam_server1
PsTestApp(V2.0,000(	↓	AdminServer
PsTestApp(V2.0,000(	↓	Server1
PsTestApp(V2.0,000(	↓	Server2
PsTestApp(V2.0,1.1.	↓	AdminServer
PsTestApp(V2.0,1.1.	↓	Server1
PsTestApp(V2.0,1.1.	↓	Server2

Fusion Middleware

Name	Status	Host
WebLogic Domain		
soa_rc1_domain		
""	↓	
AdminServer	↑	stasa39.us.oracl
bam_server1	↑	stasa39.us.oracl
Cluster1		
Server1	↑	stasa39.us.oracl
Server2	↑	stasa39.us.oracl
ser-3	↓	
soa_server1	↓	
BAM		
OracleBamServer (ba	↑	stasa39.us.oracl
OracleBamWeb (bam	↑	stasa39.us.oracl
Metadata Repositories		
mds-FileRepos1		stasa39.us.oracl
mds-owsm		stasa39.us.oracl

Farm Resource Center

Before You Begin

- Introduction to Oracle Fusion Middleware
- Understanding Key Oracle Fusion Middleware Farm Conce
- Overview of Oracle Fusion Middleware Administration Too

Typical Administration Tasks

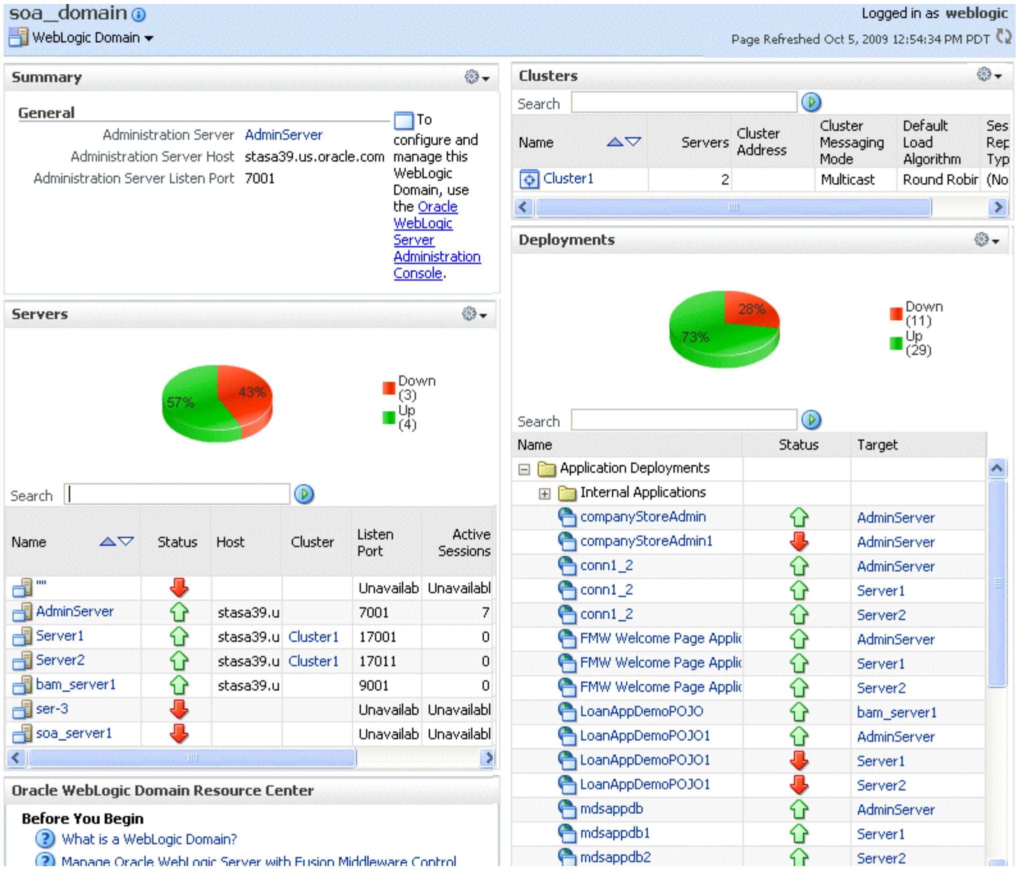
- Getting Started Using Oracle Enterprise Manager Fusion M

10.1.2 Monitoring an Oracle WebLogic Server Domain

You can view the status of a domain, including the servers, clusters, and deployments in the domain in the domain home page of Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **WebLogic Domain.**
2. Select the domain.

   The domain home page is displayed, as shown in the following figure:



soa_domain

WebLogic Domain

Logged in as **weblogic**

Page Refreshed Oct 5, 2009 12:54:34 PM PDT

Summary

General

Administration Server	AdminServer
Administration Server Host	stasa39.us.oracle.com
Administration Server Listen Port	7001

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

Servers

57% 43%

🟥 Down (3)

🟩 Up (4)

Search

Name	Status	Host	Cluster	Listen Port	Active Sessions
""	↓			Unavailab	Unavailabl
AdminServer	↑	stasa39.u		7001	7
Server1	↑	stasa39.u	Cluster1	17001	0
Server2	↑	stasa39.u	Cluster1	17011	0
bam_server1	↑	stasa39.u		9001	0
ser-3	↓			Unavailab	Unavailabl
soa_server1	↓			Unavailab	Unavailabl

Oracle WebLogic Domain Resource Center

Before You Begin

What is a WebLogic Domain?

Manage Oracle WebLogic Server with Fusion Middleware Control

Clusters

Search

Name	Servers	Cluster Address	Cluster Messaging Mode	Default Load Algorithm	Ses Rep Typ
Cluster1	2		Multicast	Round Robir	(No

Deployments

28% 73%

🟥 Down (11)

🟩 Up (29)

Search

Name	Status	Target
Application Deployments		
Internal Applications		
companyStoreAdmin	↑	AdminServer
companyStoreAdmin1	↓	AdminServer
conn1_2	↑	AdminServer
conn1_2	↑	Server1
conn1_2	↑	Server2
FMW Welcome Page Applic	↑	AdminServer
FMW Welcome Page Applic	↑	Server1
FMW Welcome Page Applic	↑	Server2
LoanAppDemoPOJO	↑	bam_server1
LoanAppDemoPOJO1	↑	AdminServer
LoanAppDemoPOJO1	↓	Server1
LoanAppDemoPOJO1	↓	Server2
mdsappdb	↑	AdminServer
mdsappdb1	↑	Server1
mdsappdb2	↑	Server2

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

> **See Also:** "Overview of the Administration Console" in the *Oracle Fusion Middleware Introduction to Oracle WebLogic Server* for information about monitoring an Oracle WebLogic Server domain using the Oracle WebLogic Server Administration Console. The Administration Console provides details about the health and performance of the domain.

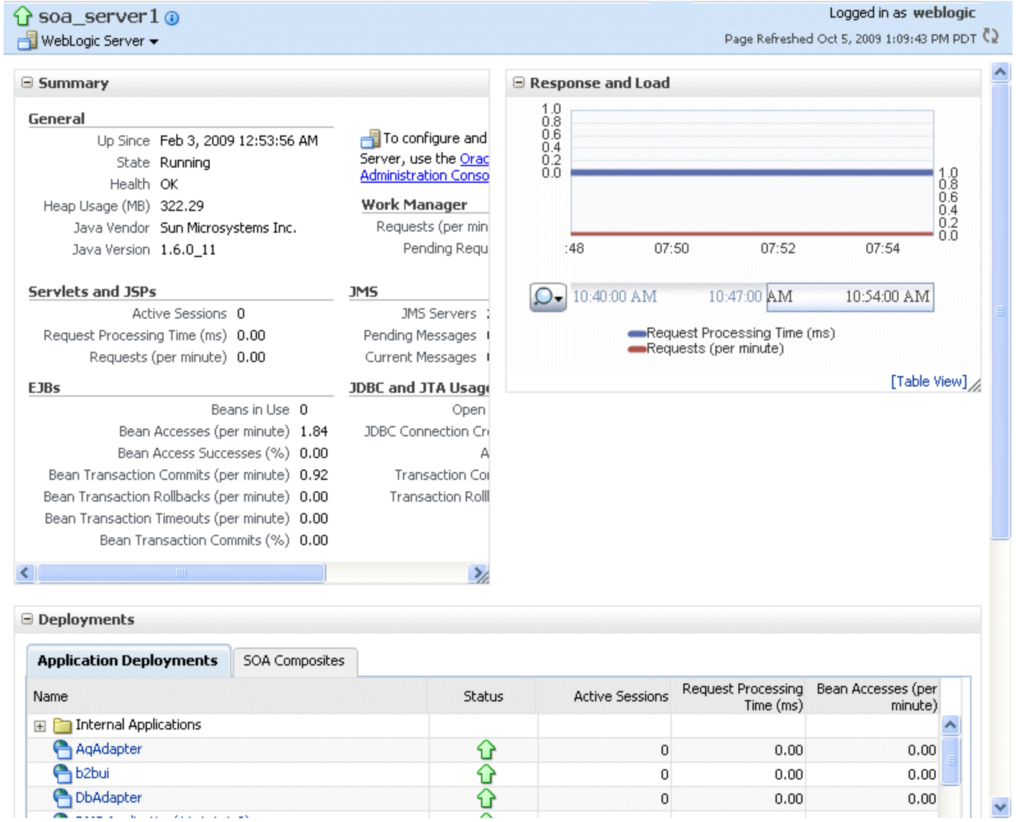
10.1.3 Monitoring an Oracle WebLogic Server Administration or Managed Server

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 farm, then **WebLogic Domain**, and then the domain.
2. Select the server.

   The server home page is displayed.

The following figure shows the home page for a Managed Server:



soa_server1

WebLogic Server ▾

Logged in as weblogic

Page Refreshed Oct 5, 2009 1:09:43 PM PDT

Summary

General

Up Since	Feb 3, 2009 12:53:56 AM
State	Running
Health	OK
Heap Usage (MB)	322.29
Java Vendor	Sun Microsystems Inc.
Java Version	1.6.0_11

To configure and Server, use the Orac Administration Conso

Work Manager

Requests (per min

Pending Requ

Servlets and JSPs

Active Sessions	0
Request Processing Time (ms)	0.00
Requests (per minute)	0.00

JMS

JMS Servers

Pending Messages

Current Messages

EJBs

Beans in Use	0
Bean Accesses (per minute)	1.84
Bean Access Successes (%)	0.00
Bean Transaction Commits (per minute)	0.92
Bean Transaction Rollbacks (per minute)	0.00
Bean Transaction Timeouts (per minute)	0.00
Bean Transaction Commits (%)	0.00

JDBC and JTA Usage

Open

JDBC Connection Cr

A

Transaction Co

Transaction Roll

Response and Load

Deployments

Application Deployments | SOA Composites

Name	Status	Active Sessions	Request Processing Time (ms)	Bean Accesses (per minute)
⊞ Internal Applications				
AqAdapter	⇧	0	0.00	0.00
b2bui	⇧	0	0.00	0.00
DbAdapter	⇧	0	0.00	0.00

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

> **See Also:** "Overview of the Administration Console" in the *Oracle Fusion Middleware Introduction to Oracle WebLogic Server* for information about monitoring servers using the Oracle WebLogic Server Administration Console. The Administration Console provides details about the health and performance of the server.

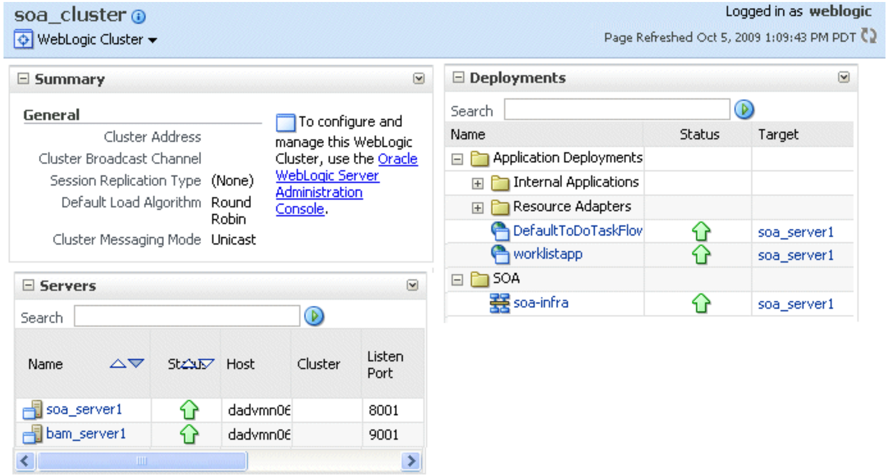
10.1.4 Monitoring a Cluster

You can view the status of a cluster, including the servers and deployments in the cluster using Fusion Middleware Control.

To monitor a cluster:

1. From the navigation pane, expand the farm, then **WebLogic Domain**, and then the domain.
2. Select the cluster.

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



soa_cluster

WebLogic Cluster

Logged in as **weblogic**

Page Refreshed Oct 5, 2009 1:09:43 PM PDT

Summary

General

Cluster Address	
Cluster Broadcast Channel	
Session Replication Type	(None)
Default Load Algorithm	Round Robin
Cluster Messaging Mode	Unicast

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

Servers

Search

Name	Status	Host	Cluster	Listen Port
soa_server1	⇧	dadvmn06		8001
bam_server1	⇧	dadvmn06		9001

Deployments

Search

Name	Status	Target
Application Deployments		
Internal Applications		
Resource Adapters		
DefaultToDoTaskFlov	⇧	soa_server1
worklistapp	⇧	soa_server1
SOA		
soa-infra	⇧	soa_server1

This page shows the following:

- A general summary of the cluster, including broadcast channel, if appropriate, the load algorithm and the messaging mode.
- A server section, with a table listing the servers that are part of the cluster.
- A deployments section with information about the applications deployed to the cluster.

> **See Also:** "Overview of the Administration Console" in the *Oracle Fusion Middleware Introduction to 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.

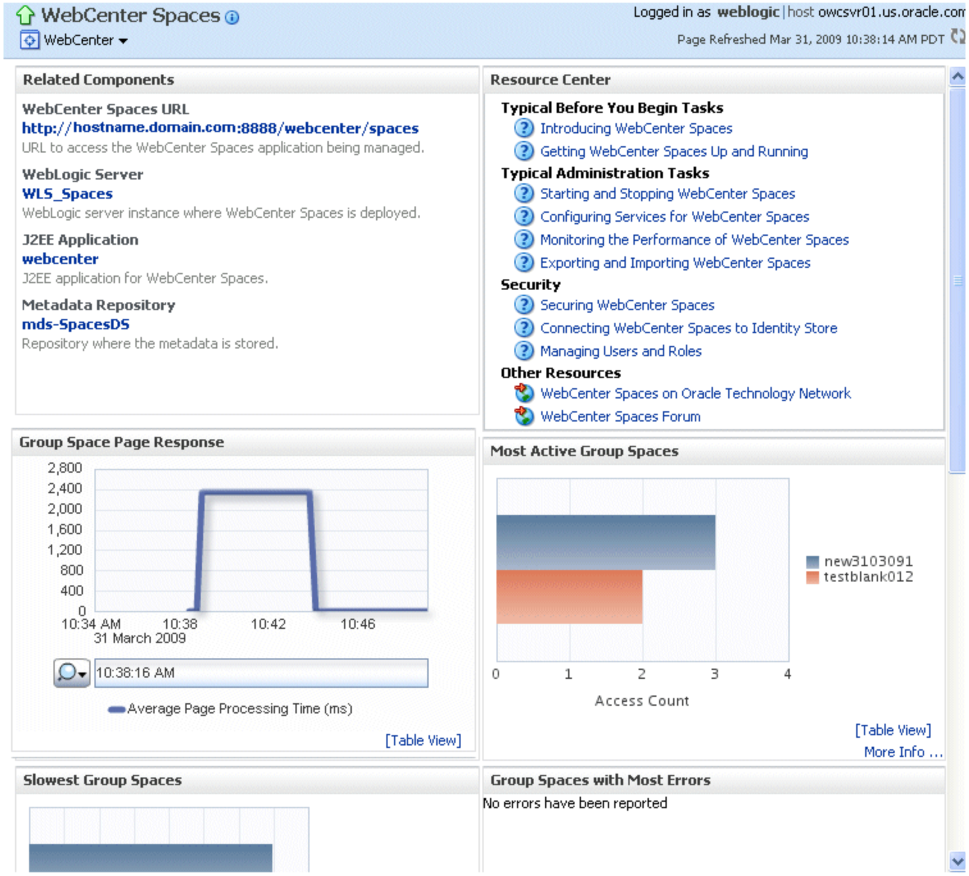
10.1.5 Monitoring a Component

You can view the status of a component, including whether the component is started, in the component home page in Fusion Middleware Control.

To monitor a Java component, such as WebCenter Spaces:

1. From the navigation pane, expand the farm, then the type of component, such as WebCenter, then the component, such as WebCenter Spaces.
2. Select the component. For example, select **WebCenter Spaces.**

   The component home page is displayed, as shown in the following figure:



WebCenter Spaces

WebCenter

Logged in as weblogic | host owcsvr01.us.oracle.com

Page Refreshed Mar 31, 2009 10:38:14 AM PDT

Related Components

WebCenter Spaces URL
http://hostname.domain.com:8888/webcenter/spaces
URL to access the WebCenter Spaces application being managed.

WebLogic Server
WLS_Spaces
WebLogic server instance where WebCenter Spaces is deployed.

J2EE Application
webcenter
J2EE application for WebCenter Spaces.

Metadata Repository
mds-SpacesDS
Repository where the metadata is stored.

Resource Center

Typical Before You Begin Tasks
- Introducing WebCenter Spaces
- Getting WebCenter Spaces Up and Running

Typical Administration Tasks
- Starting and Stopping WebCenter Spaces
- Configuring Services for WebCenter Spaces
- Monitoring the Performance of WebCenter Spaces
- Exporting and Importing WebCenter Spaces

Security
- Securing WebCenter Spaces
- Connecting WebCenter Spaces to Identity Store
- Managing Users and Roles

Other Resources
- WebCenter Spaces on Oracle Technology Network
- WebCenter Spaces Forum

Group Space Page Response

[Table View]

Most Active Group Spaces

[Table View]
More Info ...

Slowest Group Spaces

Group Spaces with Most Errors

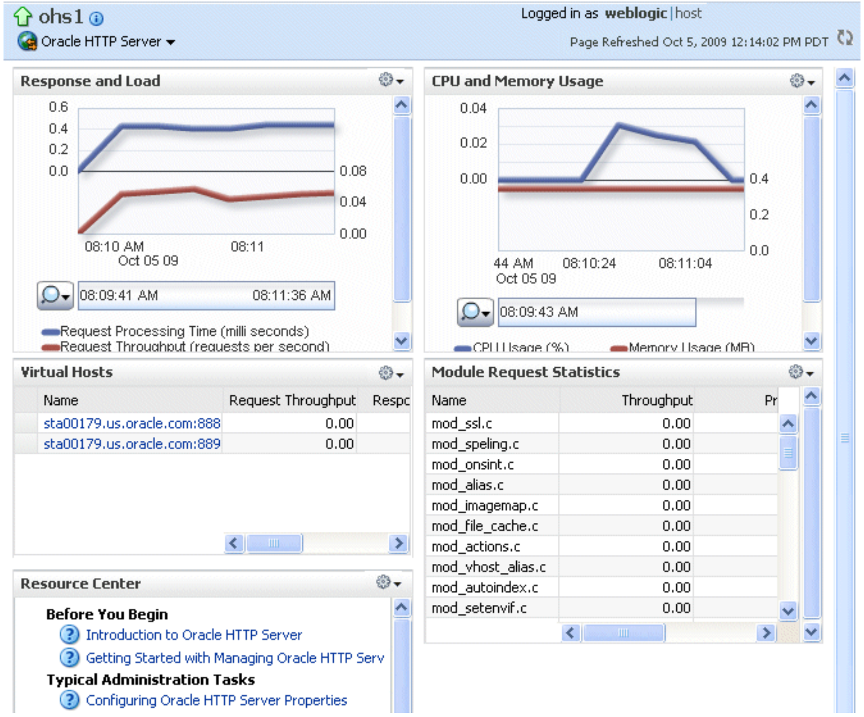
No errors have been reported

See Also: "Overview of the Administration Console" in the *Oracle Fusion Middleware Introduction to Oracle WebLogic Server* for information about using the Oracle WebLogic Server Administration Console to monitor Java components.

To monitor system components, such as Oracle HTTP Server:

1. From the navigation pane, expand the farm, then **Web Tier.**
2. Select the component, such as ohs1.

   The component home page is displayed.



ohs1

Oracle HTTP Server

Logged in as **weblogic** | host

Page Refreshed Oct 5, 2009 12:14:02 PM PDT

Response and Load

CPU and Memory Usage

Virtual Hosts

Name	Request Throughput	Respc
sta00179.us.oracle.com:888	0.00	
sta00179.us.oracle.com:889	0.00	

Module Request Statistics

Name	Throughput	Pr
mod_ssl.c	0.00	
mod_speling.c	0.00	
mod_onsint.c	0.00	
mod_alias.c	0.00	
mod_imagemap.c	0.00	
mod_file_cache.c	0.00	
mod_actions.c	0.00	
mod_vhost_alias.c	0.00	
mod_autoindex.c	0.00	
mod_setenvif.c	0.00	

Resource Center

Before You Begin

- Introduction to Oracle HTTP Server
- Getting Started with Managing Oracle HTTP Serv

Typical Administration Tasks

- Configuring Oracle HTTP Server Properties

10.1.6 Monitoring Java EE Applications

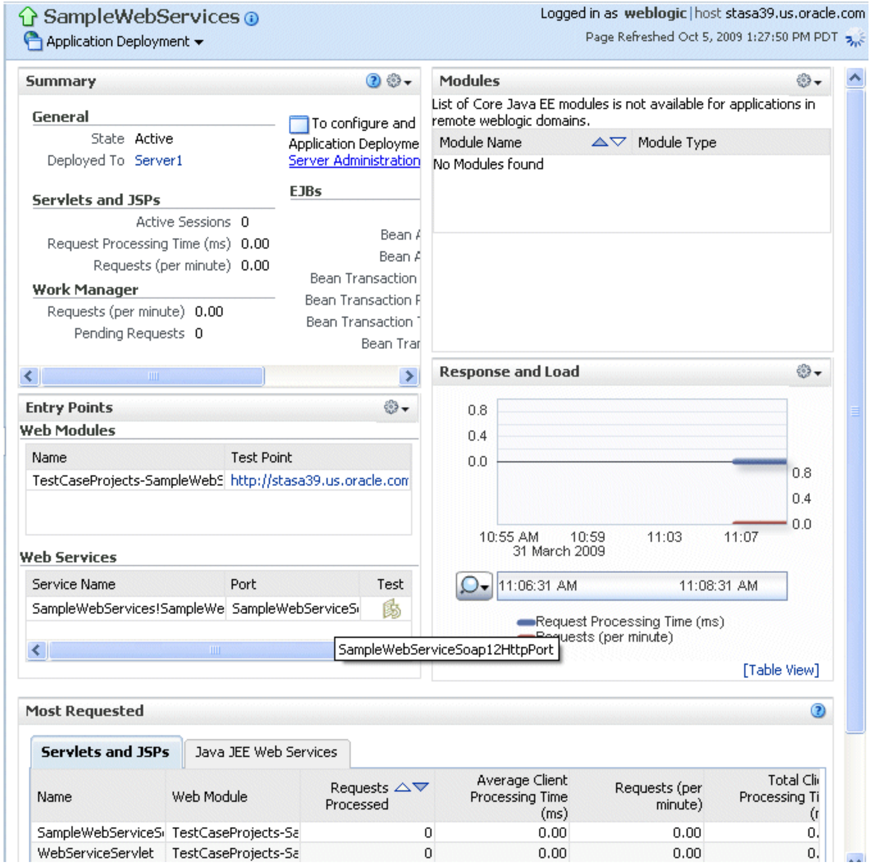
To monitor a Java EE application using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments,** then select the application to monitor.

   The application's home page is displayed.

2. 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:



SampleWebServices

Application Deployment ▾

Logged in as **weblogic** | host stasa39.us.oracle.com

Page Refreshed Oct 5, 2009 1:27:50 PM PDT

Summary

General

State Active

Deployed To Server1

□ To configure and Application Deployme Server Administration

Servlets and JSPs

Active Sessions 0

Request Processing Time (ms) 0.00

Requests (per minute) 0.00

EJBs

Bean A

Bean A

Bean Transaction

Bean Transaction F

Bean Transaction T

Bean Trar

Work Manager

Requests (per minute) 0.00

Pending Requests 0

Modules

List of Core Java EE modules is not available for applications in remote weblogic domains.

Module Name	Module Type
No Modules found	

Entry Points

Web Modules

Name	Test Point
TestCaseProjects-SampleWebS	http://stasa39.us.oracle.com

Web Services

Service Name	Port	Test
SampleWebServices!SampleWe	SampleWebServiceS	

SampleWebServiceSoap12HttpPort

Response and Load

Most Requested

Servlets and JSPs | Java JEE Web Services

Name	Web Module	Requests Processed	Average Client Processing Time (ms)	Requests (per minute)	Total Cli Processing Ti (r
SampleWebServiceS	TestCaseProjects-Sa	0	0.00	0.00	0.
WebServiceServlet	TestCaseProjects-Sa	0	0.00	0.00	0.

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
- Entry points, including any Web modules and Web services
- A list of modules with the type of module for each
- Response and load, which shows the requests per second and the request processing time
- A list of most requested servlets and JSPs

10.1.7 Monitoring ADF Applications

To monitor an ADF application:

1. From the navigation pane, expand **Application Deployments,** then select the application to monitor.

   The application's home page is displayed.

2. 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.

10.1.8 Monitoring SOA Composite Applications

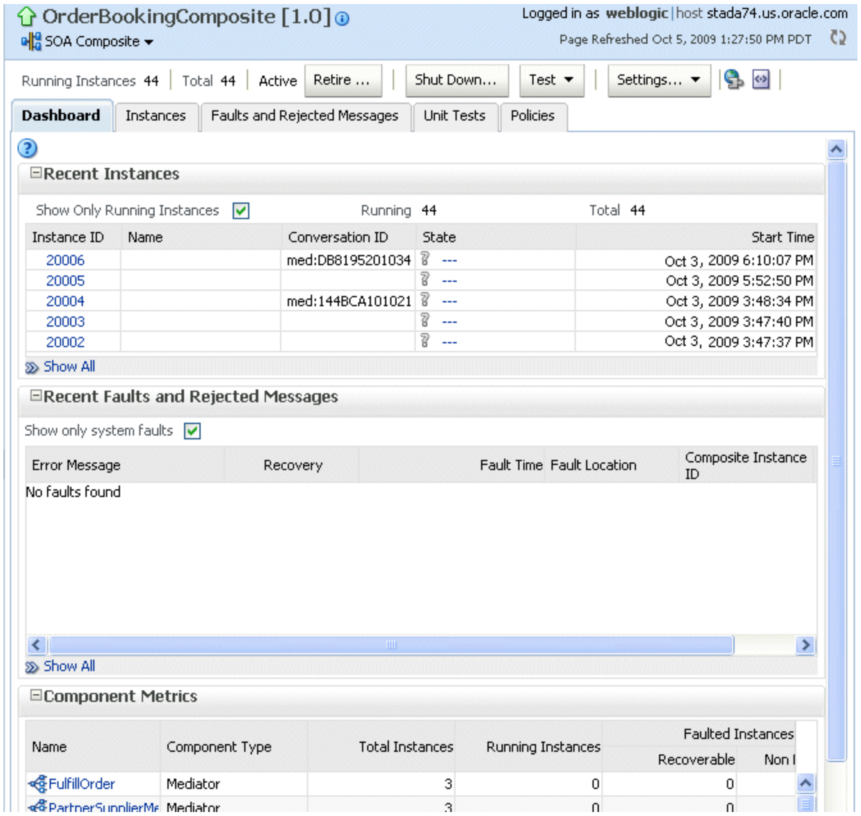
To monitor a SOA composite application:

1. From the navigation pane, expand **SOA,** then **soa-infra.** Select the application to monitor.

   The application's home page is displayed.

2. From this page, you can monitor the running instances, faults and rejected messages, and component metrics.

   The following figure shows part of a SOA composite home page:



OrderBookingComposite [1.0]

Logged in as **weblogic** | host stada74.us.oracle.com

Page Refreshed Oct 5, 2009 1:27:50 PM PDT

Running Instances 44 | Total 44 | Active

Recent Instances

Show Only Running Instances ☑ Running 44 Total 44

Instance ID	Name	Conversation ID	State	Start Time
20006		med:DB8195201034	---	Oct 3, 2009 6:10:07 PM
20005			---	Oct 3, 2009 5:52:50 PM
20004		med:144BCA101021	---	Oct 3, 2009 3:48:34 PM
20003			---	Oct 3, 2009 3:47:40 PM
20002			---	Oct 3, 2009 3:47:37 PM

Recent Faults and Rejected Messages

Show only system faults ☑

Error Message	Recovery	Fault Time	Fault Location	Composite Instance ID
No faults found				

Component Metrics

Name	Component Type	Total Instances	Running Instances	Faulted Instances	
				Recoverable	Non I
FulfillOrder	Mediator	3	0	0	
PartnerSupplierMe	Mediator	3	0	0	

This page, with the Dashboard tab selected, shows the following:

- The recent instances
- Recent faults and rejected messages
- Component metrics

10.1.9 Monitoring Oracle WebCenter Applications

To monitor an Oracle WebCenter application:

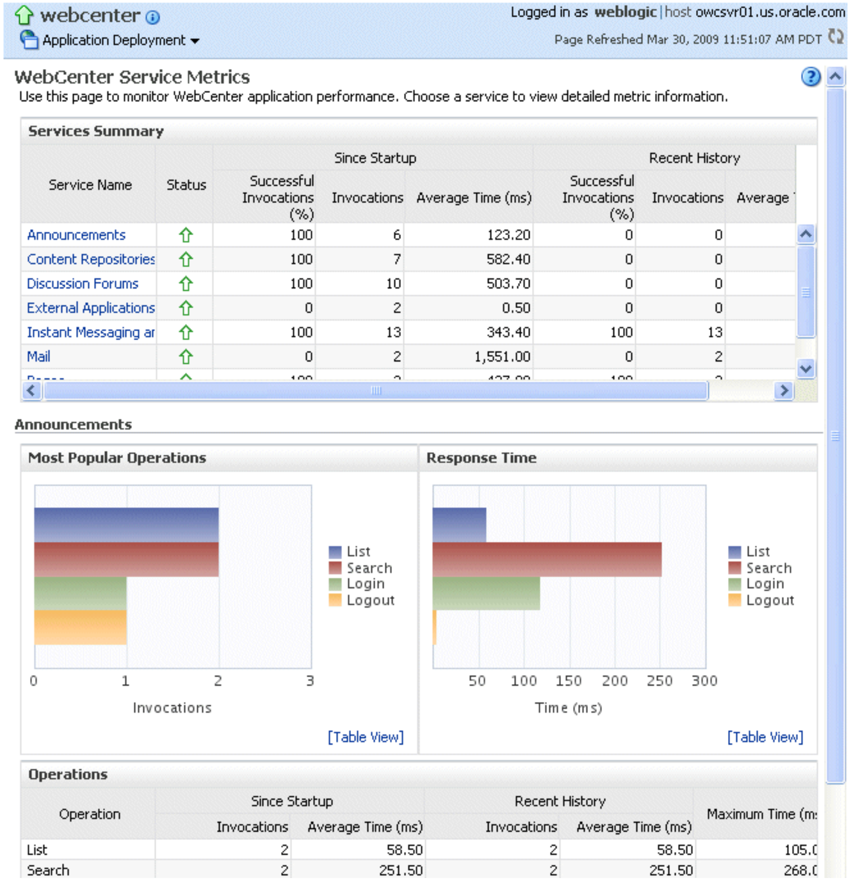
1. From the navigation pane, expand **Application Deployments,** then select the application to monitor.

   The application's home page is displayed.

2. 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.

3. To view service metrics, from the Application Deployment menu, choose **Web Center,** then **Service Metrics.**

   The following figure shows the Service Metrics page:



webcenter
Application Deployment

Logged in as **weblogic** | host owcsvr01.us.oracle.com
Page Refreshed Mar 30, 2009 11:51:07 AM PDT

WebCenter Service Metrics

Use this page to monitor WebCenter application performance. Choose a service to view detailed metric information.

Services Summary

Service Name	Status	Since Startup			Recent History		
		Successful Invocations (%)	Invocations	Average Time (ms)	Successful Invocations (%)	Invocations	Average
Announcements	⇧	100	6	123.20	0	0	
Content Repositories	⇧	100	7	582.40	0	0	
Discussion Forums	⇧	100	10	503.70	0	0	
External Applications	⇧	0	2	0.50	0	0	
Instant Messaging ar	⇧	100	13	343.40	100	13	
Mail	⇧	0	2	1,551.00	0	2	

Announcements

Most Popular Operations

Response Time

Operations

Operation	Since Startup		Recent History		Maximum Time (m:
	Invocations	Average Time (ms)	Invocations	Average Time (ms)	
List	2	58.50	2	58.50	105.(
Search	2	251.50	2	251.50	268.(

See Also: "Understanding WebCenter Performance Metrics" in *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter* for more information about service metrics

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 10*g* Grid Control.

For example, to view the performance of an Oracle WebLogic Server Managed Server:

1. From the navigation pane, expand the farm, then **WebLogic Domain**, and then the domain.

2. Select the server to monitor.

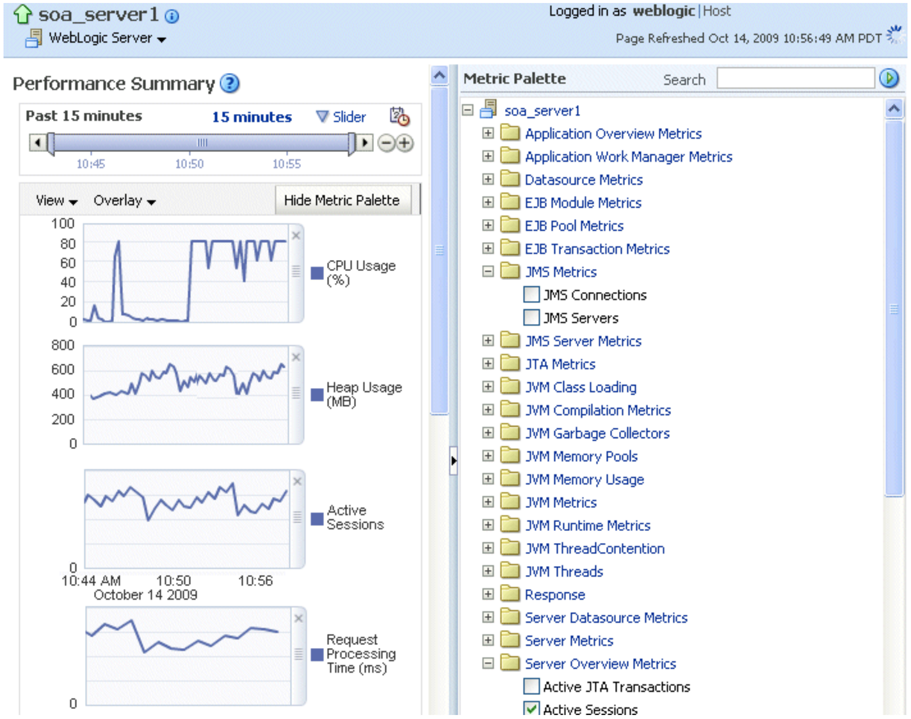
   The Managed Server home page is displayed.

3. From the WebLogic Server menu, choose **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 additional metrics to add them 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.**

You can also view the performance of a components, such as Oracle HTTP Server or Oracle SOA Suite. 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 farm. The Topology Viewer is a graphical representation of routing relationships across components and elements of the farm. You can easily determine how requests are routed across components. For example, you can see how requests are routed from Oracle Web Cache, to Oracle HTTP Server, to a Managed Server, to a data source.

Note: To view relationships between Oracle WebLogic Server, Oracle Web Cache, 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 or save it to a .png file.

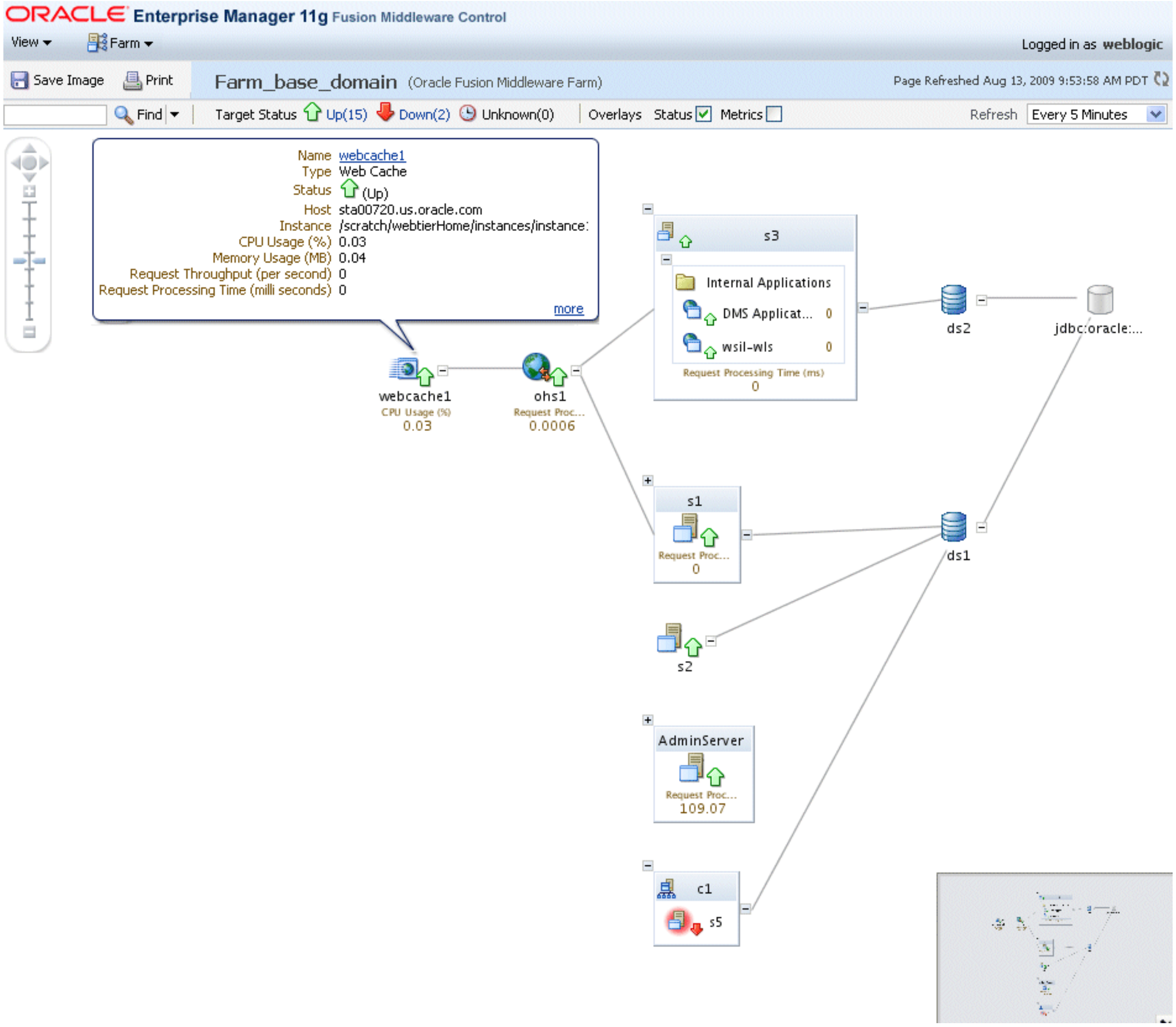
To view the topology using Fusion Middleware Control:

1. Click **Topology.**

   The Topology Viewer is displayed in a separate window.

2. To see information about a particular target, place your mouse over the target. To view additional information, click **More.**

   The following shows the Topology Viewer window, with information about the Oracle Web Cache component webcache1:



Farm_base_domain (Oracle Fusion Middleware Farm)

Page Refreshed Aug 13, 2009 9:53:58 AM PDT

Target Status Up(15) Down(2) Unknown(0)

Overlays Status ☑ Metrics □

Refresh Every 5 Minutes

Name webcache1
Type Web Cache
Status (Up)
Host sta00720.us.oracle.com
Instance /scratch/webtierHome/instances/instance
CPU Usage (%) 0.03
Memory Usage (MB) 0.04
Request Throughput (per second) 0
Request Processing Time (milli seconds) 0

3. From the View menu, you can save or print the image, expand or collapse all of the nodes, or change the orientation of the topology to be left to right or top to bottom.

   In addition, you can refresh the status and the metrics or update the topology. To refresh the status and metrics, click **Refresh Target Status and Metrics.** To update the topology shown in the viewer, click **Update Topology.** If a target has been added or deleted, the target list and relationships are updated. This option also updates the status and metrics.

4. From the **Refresh** dropdown, you can refresh manually, or you can enable automatically refreshing the status and metrics, every minute, every five minutes, or every thirty minutes. By default, the Topology Viewer refreshes the metrics every 5 minutes.

5. With Topology Viewer, you can also:

   - Search for a target within the topology. This makes it easier to find a target if you have many targets. Enter the name in the **Find** box. The target is highlighted and 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.

   - View the targets by status. Choose **Up, Down,** or **Unknown** from the Target Status at the top of the page.

   - Navigate to the home page of a target. Right-click the target, and select **Home.**

   - Hide or show the status or metrics. Click **Status** or **Metrics** in the Overlays section.

     If you select Metrics, one key performance metric for the component is displayed. (You cannot change the metric that is displayed.)

   - View the routing relationships between components. For example, you can view the routing from Oracle Web Cache to Oracle HTTP Server to Oracle WebLogic Server.

   - You can 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.

6. To change what is visible in the topology view, drag the shaded section in the navigator window, which is located in the bottom right.

Notes:

- If you use Mozilla Firefox, when you click an entity in Topology Viewer to take you back to the main Fusion Middleware Control window, focus is not returned to the main window. For example, if you right-click an entity and select logs from menu, the focus remains on the Topology Viewer window. (If you go back to the main window, the Logs page is correctly displayed.)

  To workaround this problem, make the following change in Firefox:

  From the Tools menu, select **Options,** and then **Content.** Click **Advanced.** In the Advanced JavaScript Settings dialog box, select **Raise and lower windows.**

- If you use Internet Explorer, turn off the **Always Open Popups in New Tab** option.

11

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, to view and manage log files to assist in monitoring system activity, and to diagnose problems.

It contains the following topics:

11.1 Overview of Oracle Fusion Middleware 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 and, 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.
- 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 *Oracle Fusion Middleware 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 SOA Suite:

```
[2009-10-23T10:54:00.206-07:00] [soa_server1] [NOTIFICATION] [] [oracle.mds] [tid:
[STANDBY].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)']
[userId: <anonymous>] [ecid: 0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0] [APP: wsm-pm]
"Metadata Services: Metadata archive (MAR) not found."
```

In the message, the fields map to the following attributes, which are described in Table 11–1:

- `2009-09-23T10:54:00.206-07:00`: Timestamp, originating
- `soa_server1`: Organization ID
- `NOTIFICATION`: Message Type
- `oracle.mds`: Component ID
- `tid: [STANDBY].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'`: Thread ID
- `[userId: <anonymous>]: User ID`
- `ecid: 0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0`: Execution Context ID
- `APP: wsm-pm`: Supplemental Attribute
- `"Metadata Services: Metadata archive (MAR) not found."`: Message Text

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. The ID is `oracle` for all Oracle components.	No
INSTANCE_ID (INST_ID)	The name of the Oracle instance to which the component that originated the message belongs.	No
COMPONENT ID (COMP)	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

Table 11–1 (Cont.) ODL Format Message Fields

Attribute Name	Description	Required
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
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.	No
MESSAGE TEXT (TEXT)	The text of the error 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

For most Java components, the log file location is:

```
(UNIX) MW_HOME/user_projects/domains/domain_name/servers/server_name/logs
(Windows) MW_HOME\user_projects\domains\domain_name\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) ORACLE_INSTANCE/diagnostics/logs
(Windows) ORACLE_INSTANCE\diagnostics\logs
```

Table 11–2 shows the log file location for components of Oracle Fusion Middleware.

In the table, *DOMAIN_HOME* refers to the following directory, which is the WebLogic Server domain home:

```
MW_HOME/user_projects/domains/domain_name
```

In the table, *ORACLE_INSTANCE* refers to the following directory, which is the Oracle instance home:

```
MW_HOME/instance_name
```

Table 11–2 Log File Location

Component	Log File Location
Oracle Application Development Framework	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Business Activity Monitoring	*DOMAIN_HOME*/servers/*server_name*/logs/bam-diagnostic.log
Oracle Business Intelligence Discoverer	*DOMAIN_HOME*/servers/*server_name*/logs/discoverer/server/diagnostic.log *DOMAIN_HOME*/servers/*server_name*/logs/discoverer/*server_name*-diagnostic.log *DOMAIN_HOME*/servers/*server_name*/logs/discoverer/diagnostic.log
Oracle Directory Integration Platform	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Forms Services	*MW_HOME*/user_projects/domains/*domain_name*/servers/*server_name*/logs/*server-name*-diagnostic.log *ORACLE_HOME*/j2ee/DevSuite/application-deployments/forms/application.log
Oracle Fusion Middleware Audit Framework	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Fusion Middleware Control	*DOMAIN_HOME*/sysman/log/emoms.log *DOMAIN_HOME*/sysman/log/emoms.trc
Oracle HTTP Server	*ORACLE_INSTANCE*/diagnostics/logs/OHS/*component_name*/*.log
Oracle Identity Federation	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Internet Directory	*ORACLE_INSTANCE*/diagnostics/logs/OID/oid*.log *ORACLE_INSTANCE*/diagnostics/logs/OID/tools/*.log
Oracle Platform Security Services	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Portal	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Reports	*ORACLE_INSTANCE*/diagnostics/logs/ReportsServerComponent *ORACLE_INSTANCE*/diagnostics/logs/ReportsBridgeComponent *ORACLE_INSTANCE*/diagnostics/logs/ReportsToolsComponent
Oracle Virtual Directory	*ORACLE_INSTANCE*/config/OVD/*component_name*/diagnostic.log *ORACLE_INSTANCE*/diagnostics/logs/OVD/*component_name*/diagnostic.log
Repository Creation Utility	By default, writes to file specified in RCU_LOG_LOCATION. If not specified, attempts to write to the following locations: **1.** *ORACLE_HOME*/rcu/log/*timestamp* **2.** /tmp/logdir.*timestamp*
Oracle TopLink	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle SOA Suite	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Web Cache	*ORACLE_INSTANCE*/diagnostics/logs/WebCache/*component_name**-log
Oracle WebCenter	*DOMAIN_HOME*/servers/*server_name*/logs/*component*-diagnostic.log
Oracle WebLogic Server	*DOMAIN_HOME*/servers/*server_name*/logs/*server_name*-diagnostic.log
Oracle Web Services Manager	*DOMAIN_HOME*/servers/*server_name*/logs/owsm/msglogging *DOMAIN_HOME*/servers/*server_name*/logs/owsm-diagnostic.log

11.3 Searching and Viewing Log Files

You can search, view, and list log files across Oracle Fusion Middleware components. You can search and view 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 and search log files using the WLST command-line tool.

This section covers the following topics:

- Searching Log Files
- Viewing Log Files and Their Messages
- Downloading Log Files

Note the following about using the WLST commands:

- To use the custom WLST logging commands, you must invoke the WLST script from the Oracle Common home. See Section 3.5.1.1 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 can 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.
- 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 either the WebLogic domain home, or the Oracle instance.
- 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:

  ```
  ./wlst.sh
  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.

  > **See Also:** "Logging Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

11.3.1 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

11.3.1.1 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

11.3.1.1.1 Searching Log Files: Basic Searches This section describes how to perform basic searches for log messages.

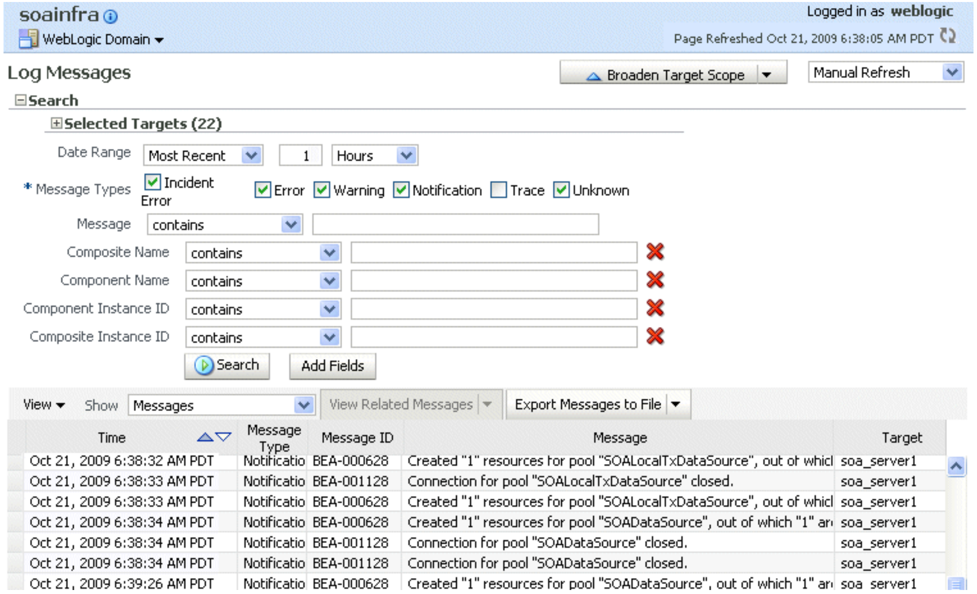
You can search for all of the messages for all of the entities in a domain, an Oracle WebLogic Server, a component, or an application.

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:



Page Refreshed Oct 21, 2009 6:38:05 AM PDT

Date Range: Most Recent 1 Hours

* Message Types: ☑ Incident Error ☑ Error ☑ Warning ☑ Notification □ Trace ☑ Unknown

Message: contains

Composite Name: contains

Component Name: contains

Component Instance ID: contains

Composite Instance ID: contains

Show: Messages

Time	Message Type	Message ID	Message	Target
Oct 21, 2009 6:38:32 AM PDT	Notificatio	BEA-000628	Created "1" resources for pool "SOALocalTxDataSource", out of whicl	soa_server1
Oct 21, 2009 6:38:33 AM PDT	Notificatio	BEA-001128	Connection for pool "SOALocalTxDataSource" closed.	soa_server1
Oct 21, 2009 6:38:33 AM PDT	Notificatio	BEA-000628	Created "1" resources for pool "SOALocalTxDataSource", out of whicl	soa_server1
Oct 21, 2009 6:38:34 AM PDT	Notificatio	BEA-000628	Created "1" resources for pool "SOADataSource", out of which "1" ar	soa_server1
Oct 21, 2009 6:38:34 AM PDT	Notificatio	BEA-001128	Connection for pool "SOADataSource" closed.	soa_server1
Oct 21, 2009 6:38:34 AM PDT	Notificatio	BEA-001128	Connection for pool "SOADataSource" closed.	soa_server1
Oct 21, 2009 6:39:26 AM PDT	Notificatio	BEA-000628	Created "1" resources for pool "SOADataSource", out of which "1" ar	soa_server1

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 11.3.1.1.2.
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 Message Type:** Summarizes the matching messages by grouping them based on message type at the target level. This is the default mode.

     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.
   - **Group by Message ID:** Summarizes the matching messages by grouping them based on message ID, message type, and module IDs at the target level.

     To see the associated messages, click the count in the **Occurrences** column. The Messages by Message ID page is displayed. To see the details of a particular message, click the message. The details are displayed below the table of messages.

11.3.1.1.2 Searching Log Files: Advanced Searches This section describes some of the advanced search mechanisms you can use.

You can refine your search criteria using the following controls in the Log Messages page:

- For **Message,** you can select an operator, such as **contains,** and 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 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.**

11.3.1.2 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.

To search for error messages generated in the last 5 minutes, for the Oracle HTTP Server ohs1, use the following command:

```
displayLogs(target='opmn:asinst_1/ohs1', last=5)
```

To search for error messages generated in the last 10 minutes for the Managed Server soa_server1, use the following command:

```
displayLogs(oracleInstance='/scratch/oracle1/Oracle/Middleware/user_
projects/domains/soa_domain', target='soa_server1', 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 soa_server1, use the following command:

  ```
  displayLogs(oracleInstance='/scratch/oracle1/Oracle/Middleware/user_
  projects/domains/soa_domain', target='soa_server1',
            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 (140.87.134.52:13934:1186078666446:0) for the Managed Server soa_server1, use the following command:

  ```
  displayLogs(oracleInstance='/scratch/oracle1/Oracle/Middleware/user_
  projects/domains/soa_domain', target='soa_server1',
           query='ecid eq 140.87.134.52:13934:1186078666446:0')
  ```

- 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:

  ```
  2007-09-30T12:00:00:0000-08:00
  ```

  For example, to display the error message from between 8:00 a.m. and 11 a.m. on April 17, 2009, use the following command:

  ```
  displayLogs(query='TSTZ_ORIGINATING from 2009-09-17T08:00:00-07:00
              and TSTZ_ORIGINATING to 2009-09-17T11:00:00-07:00')
  ```

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')
```

11.3.2 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

11.3.2.1 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.

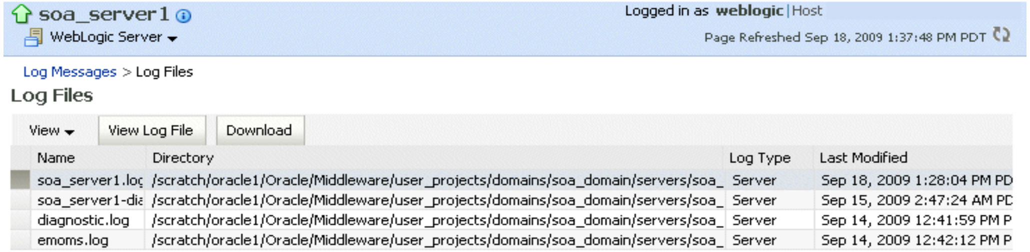
For example, to view the log files and their messages for a Managed Server:

1. From the navigation pane, expand the farm, then **WebLogic Domain**, and then the domain. Right-click the Managed Server name and choose **Logs,** then **View Log Messages.**

   The Log Messages page displays the log files for the Managed Server and any applications running in that server.

2. Expand **Selected Targets** and in the row for a particular component or application, click the **Target Log Files** icon.

   The View Log Files page is displayed. On this page, you can see a list of log files related to the Managed Server, as shown in the following figure:



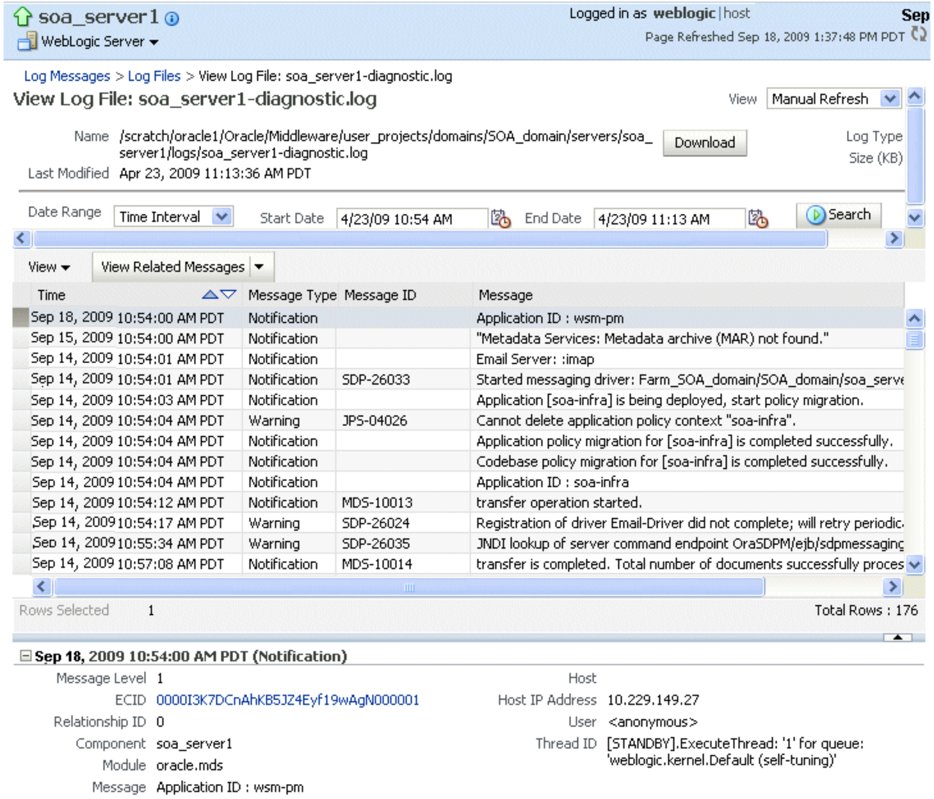
Name	Directory	Log Type	Last Modified
soa_server1.lo(	/scratch/oracle1/Oracle/Middleware/user_projects/domains/soa_domain/servers/soa_	Server	Sep 18, 2009 1:28:04 PM PD
soa_server1-di	/scratch/oracle1/Oracle/Middleware/user_projects/domains/soa_domain/servers/soa_	Server	Sep 15, 2009 2:47:24 AM PD
diagnostic.log	/scratch/oracle1/Oracle/Middleware/user_projects/domains/soa_domain/servers/soa_	Server	Sep 14, 2009 12:41:59 PM P
emoms.log	/scratch/oracle1/Oracle/Middleware/user_projects/domains/soa_domain/servers/soa_	Server	Sep 14, 2009 12:42:12 PM P

3. Select a file and click **View Log File.**

   The View Log Files page is displayed. On this page, you can view the list of messages.

4. To view the details of a message, select the message.

   The details are displayed in the pane below the listing, as shown in the following figure:



soa_server1
WebLogic Server
Logged in as weblogic|host
Sep
Page Refreshed Sep 18, 2009 1:37:48 PM PDT
Log Messages > Log Files > View Log File: soa_server1-diagnostic.log

View Log File: soa_server1-diagnostic.log

View: Manual Refresh

Name: /scratch/oracle1/Oracle/Middleware/user_projects/domains/SOA_domain/servers/soa_server1/logs/soa_server1-diagnostic.log

Download
Log Type
Size (KB)

Last Modified: Apr 23, 2009 11:13:36 AM PDT

Date Range: Time Interval   Start Date: 4/23/09 10:54 AM   End Date: 4/23/09 11:13 AM

Search
View
View Related Messages

Time	Message Type	Message ID	Message
Sep 18, 2009 10:54:00 AM PDT	Notification		Application ID : wsm-pm
Sep 15, 2009 10:54:00 AM PDT	Notification		"Metadata Services: Metadata archive (MAR) not found."
Sep 14, 2009 10:54:01 AM PDT	Notification		Email Server: :imap
Sep 14, 2009 10:54:01 AM PDT	Notification	SDP-26033	Started messaging driver: Farm_SOA_domain/SOA_domain/soa_serve
Sep 14, 2009 10:54:03 AM PDT	Notification		Application [soa-infra] is being deployed, start policy migration.
Sep 14, 2009 10:54:04 AM PDT	Warning	JPS-04026	Cannot delete application policy context "soa-infra".
Sep 14, 2009 10:54:04 AM PDT	Notification		Application policy migration for [soa-infra] is completed successfully.
Sep 14, 2009 10:54:04 AM PDT	Notification		Codebase policy migration for [soa-infra] is completed successfully.
Sep 14, 2009 10:54:04 AM PDT	Notification		Application ID : soa-infra
Sep 14, 2009 10:54:12 AM PDT	Notification	MDS-10013	transfer operation started.
Sep 14, 2009 10:54:17 AM PDT	Warning	SDP-26024	Registration of driver Email-Driver did not complete; will retry periodic
Sep 14, 2009 10:55:34 AM PDT	Warning	SDP-26035	JNDI lookup of server command endpoint OraSDPM/ejb/sdpmessaging
Sep 14, 2009 10:57:08 AM PDT	Notification	MDS-10014	transfer is completed. Total number of documents successfully proces

Rows Selected 1   Total Rows : 176

Sep 18, 2009 10:54:00 AM PDT (Notification)

Message Level: 1
ECID: 0000I3K7DCnAhKB5JZ4Eyf19wAgN000001
Relationship ID: 0
Component: soa_server1
Module: oracle.mds
Message: Application ID : wsm-pm
Host:
Host IP Address: 10.229.149.27
User: <anonymous>
Thread ID: [STANDBY].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'

By default, the messages are sorted by time, in ascending order. You can sort the messages by the columns Time, Message Type, or Message ID by clicking the column name.

5. 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.

11.3.2.2 Viewing Log Files and Their Messages Using WLST

You can list the log files for an Oracle WebLogic Server domain, server, an Oracle instance, 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 soa_server1, use the following command:

```
wls:/soa_domain/domainRuntime> listLogs(target='server_soa')
file://host/scratch/oracle1/Oracle/Middleware/user_projects/domains/SOA_
domain/servers/soa_server1/logs/soa_server1.log
2009-09-17 16:40:45                    4.9M soa_server1.log00001
2009-09-17 18:35:35                    4.9M soa_server1.log00002
2009-09-17 20:30:25                    4.9M soa_server1.log00003
...
file://host/scratch/oracle1/Oracle/Middleware/user_projects/domains/SOA_
domain/servers/soa_server1/logs/soa_server1-diagnostic.log
2009-09-22 13:53:32                     10M soa_server1-diagnostic-22.log
```

```
2009-09-22 19:18:32                     10M soa_server1-diagnostic-23.log
2009-09-23 00:42:32                     10M soa_server1-diagnostic-24.log
2009-09-23 06:07:32                     10M soa_server1-diagnostic-25.log
2009-09-23 11:31:32                     10M soa_server1-diagnostic-26.log
2009-09-23 16:56:32                     10M soa_server1-diagnostic-27.log
2009-09-23 22:20:32                     10M soa_server1-diagnostic-28.log
2009-09-24 03:45:32                     10M soa_server1-diagnostic-29.log
2009-09-24 09:11:32                     10M soa_server1-diagnostic-30.log
2009-09-24 14:08:32                    9.2M soa_server1-diagnostic.log
...
```

- To list the logs for the Oracle HTTP Server ohs1 in the Oracle instance asinst_1, use the following command:

  ```
  listLogs(target='opmn:asinst_1/ohs1')
  ```

- To list the logs while disconnected, you must specify the `oracleInstance` parameter, passing it either the Oracle WebLogic Server domain or the Oracle instance home for the system component. For example, to list the log files for the Managed Server soa_server1:

  ```
  listLogs(oracleInstance='/scratch/Oracle/Middleware/user_projects/domains/SOA_
  domain',
             target='soa_server1')
  ```

- 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)
  [2009-09-05T08:05:29.652-07:00] [soa_server1] [NOTIFICATION] [BEA-000628]
   [Common] [host: dadvmn0623] [nwaddr: 10.229.149.27] [tid:
   [ACTIVE].ExecuteThread: '10' for queue: 'weblogic.kernel.Default
   (self-tuning)'] [userId: <WLS Kernel>] [TARGET: /SOA_domain/soa_server1]
   [LOG_FILE: /scratch//Oracle/Middleware/user_projects/domains/SOA_
  domain/servers/soa_server1/logs/soa_server1.log] Created "1" resources for
   pool "SOADataSource", out of which "1" are available and "0" are unavailable.
  [2009-09-05T08:05:29.673-07:00] [soa_server1] [NOTIFICATION] [BEA-000628]
   [Common] [host: dadvmn0623] [nwaddr: 10.229.149.27] [tid:
  oracle.integration.platform.blocks.executor.WorkManagerExecutor$1@17f5105]
   [userId: <anonymous>] [TARGET: /SOA_domain/soa_server1] [LOG_FILE:
   /scratch/Oracle/Middleware/user_projects/domains/SOA
  _domain/servers/soa_server1/logs/soa_server1.log] Created "1" resources for
   pool "SOADataSource", out of which "1" are available and "0" are unavailable.
  [2009-09-05T08:05:30.448-07:00] [soa_server1] [NOTIFICATION] [BEA-001128]
   [JDBC] [host: dadvmn0623] [nwaddr: 10.229.149.27] [tid:
   oracle.integration.platform.blocks.executor.WorkManagerExecutor$1@17f5105]
   [userId: <anonymous>] [TARGET: /SOA_domain/soa_server1] [LOG_FILE:
   /scratch/Oracle/Middleware/user_projects/domains/SOA
  _domain/servers/soa_server1/logs/soa_server1.log] Connection for pool
   "SOADataSource" closed.
  ```

  The previous command returns the messages sorted by time, in ascending order.

- To display the log files for the Oracle HTTP Server ohs1 in the Oracle instance asinst_1, use the following command:

  ```
  displayLogs(target='opmn:asinst_1/ohs1')
  ```

You can search the messages by specifying particular criteria and sort the output, as described in Section 11.3.1.

> **See Also:** "Logging Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for more information about the `listLogs` and `displayLogs` commands

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, 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, select target, such as the domain.
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 11.3.1.1.
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 farm, then **WebLogic Domain,** and then 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 11.3.1.1.
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. Either select **Open With** or **Save to Disk.** Click **OK.**

To download the log files for a specific component using Fusion Middleware Control:

1. From the navigation pane, expand the farm. For system components, expand the installation type, such as **Web Tier** and select the component. For Java components, expand the farm, then 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. In the Log Files column, click a log file.

   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'], export='download_log.txt')
```

The messages are written to the file download_log.txt.

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:** Note that you cannot configure options for log files of system components, which are listed in Section 3.5.2. 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 following options for log files:

- The name and location of log files. See Section 11.4.1.
- The size of log files: You can specify that a new file is created either when the log file reaches a certain size or when a particular time is reached. This is called **log file rotation.** See Section 11.4.2.
- The level of information written to log files. See Section 11.4.3.
- The format of the log files. See Section 11.4.4.

- The Locale encoding. See Section 11.4.5.

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

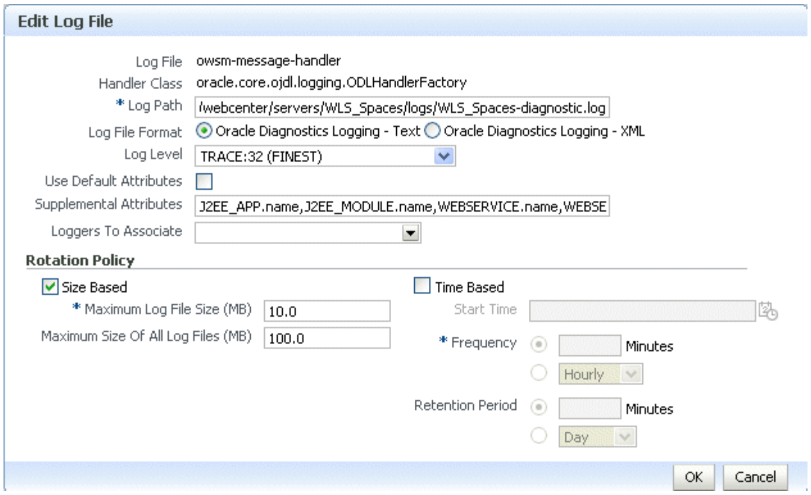
To change the location of a component's log file using Fusion Middleware Control, navigate to the component's home page and choose **Logs,** then **Log Configuration** from the dynamic target menu.

For example, 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.
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:



Edit Log File

Log File: owsm-message-handler
Handler Class: oracle.core.ojdl.logging.ODLHandlerFactory
* Log Path: /webcenter/servers/WLS_Spaces/logs/WLS_Spaces-diagnostic.log
Log File Format: ● Oracle Diagnostics Logging - Text ○ Oracle Diagnostics Logging - XML
Log Level: TRACE:32 (FINEST)
Use Default Attributes: □
Supplemental Attributes: J2EE_APP.name,J2EE_MODULE.name,WEBSERVICE.name,WEBSE
Loggers To Associate:

Rotation Policy

☑ Size Based
* Maximum Log File Size (MB): 10.0
Maximum Size Of All Log Files (MB): 100.0

□ Time Based
Start Time:
* Frequency: ● Minutes ○ Hourly
Retention Period: ● Minutes ○ Day

5. For **Log Path,** enter a new path.
6. Click **OK.**
7. In the confirmation window, click **Close.**

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, for the log files of some components, 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 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.
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,** enter the date when you want the rotation to start. For example, enter 10-SEP-2009.
- 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 5MB 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 message level are from 1 (highest severity) through 32 (lowest severity). Some components support only some of the levels for each message type. 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 shows 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.

Table 11–3 (Cont.) Diagnostic Message Types and Level

Message Type	Level	Description
WARNING	1	A potential problem that should be reviewed by the administrator. Examples are invalid parameter values or a specified file does not exist.
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

Table 11–4 (Cont.) Mapping of Log Levels Among ODL, Oracle WebLogic Server, and

ODL	WebLogic Server	Java
NOTIFICATION:16	(DEBUG)	700 - CONFIG
TRACE:1	(DEBUG)	500 - FINE
TRACE:1	DEBUG	495
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

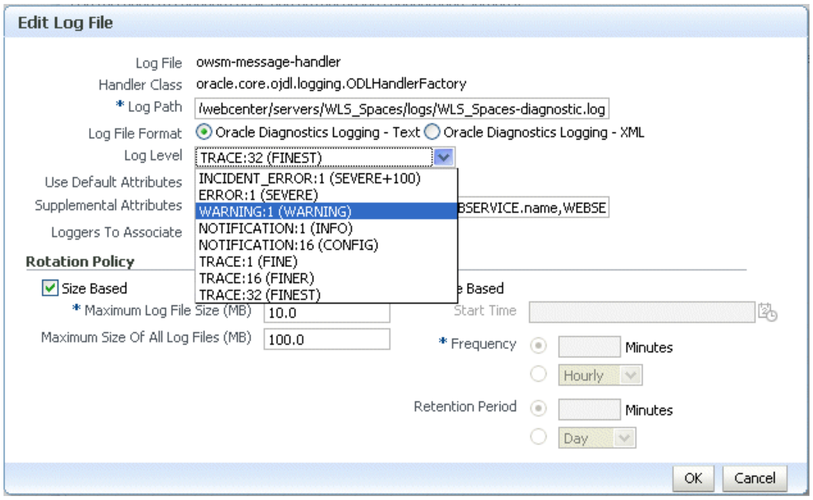
You can set the message level for a particular log file or for loggers.

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.
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:



Edit Log File

Log File owsm-message-handler

Handler Class oracle.core.ojdl.logging.ODLHandlerFactory

* Log Path /webcenter/servers/WLS_Spaces/logs/WLS_Spaces-diagnostic.log

Log File Format ● Oracle Diagnostics Logging - Text ○ Oracle Diagnostics Logging - XML

Log Level TRACE:32 (FINEST)

Supplemental Attributes BSERVICE.name,WEBSE

☑ Size Based

* Maximum Log File Size (MB) 10.0

Maximum Size Of All Log Files (MB) 100.0

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.
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 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 **Root Logger** for run-time loggers or **oracle** for persistent loggers. Then, for the child loggers, specify **Inherit from Parent.** For most situations, that is sufficient.

   However, if you need to specify the level for a particular logger, expand **Root Logger** or **oracle,** 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 soa_server1, use the following command:

```
getLogLevel(logger='oracle',  target='soa_server1')
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.soa:

```
setLogLevel(target='soa_server1', logger='oracle.soa', level='WARNING')
```

To get a list of loggers for the Oracle WebLogic Server soa_server1, use the listLoggers command:

```
listLoggers(target='soa_server1')
.
.
.
oracle.soa                              |  WARNING:1
oracle.soa.adapter                      | <Inherited>
orac                                    | <Inherited>
oracle.soa.b2b.apptransport             | <Inherited>
oracle.soa.b2b.engine                   | <Inherited>
oracle.soa.b2b.repository               | <Inherited>
oracle.soa.b2b.transport                | <Inherited>
oracle.soa.b2b.ui                       | <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 soa_server1, use the following command:

```
listLoggers(target='soa_server1', pattern='oracle.*')
oracle                                     | NOTIFICATION:1
oracle.adapter                             | <Inherited>
oracle.adapter.jms.logger                  | <Inherited>
oracle.adf                                 | <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.
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 - 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 use the `configureLogHandler` command. You use 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, 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). On Windows, you can enable UTF-8 logging as described in the following topics.

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. You can specify the encoding property to specify the character set encoding.

The logging.xml file is located in the following directory:

```
DOMAIN_HOME/config/fmwconfig/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:

```
152.68.202.244:43750:1172674368694:1
```

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. For example:

```
displayLogs(ecid='0000Hl9TwKUCslT6uBi8UH18lkWX000002')
```

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.**

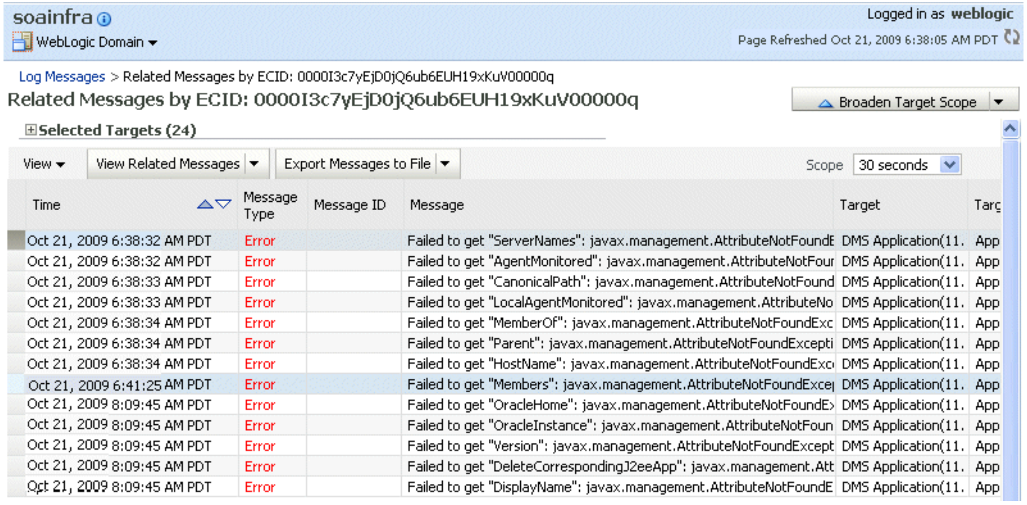
   To search for messages for a component or application, select the component or application and then choose **Logs,** then **View Log Messages** from that target's menu.

2. Specify search criteria, as described in Section 11.3.1.1.2.

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:



Related Messages by ECID: 0000I3c7yEjD0jQ6ub6EUH19xKuV00000q

Time	Message Type	Message ID	Message	Target	Tarç
Oct 21, 2009 6:38:32 AM PDT	Error		Failed to get "ServerNames": javax.management.AttributeNotFoundE	DMS Application(11.	App
Oct 21, 2009 6:38:32 AM PDT	Error		Failed to get "AgentMonitored": javax.management.AttributeNotFour	DMS Application(11.	App
Oct 21, 2009 6:38:33 AM PDT	Error		Failed to get "CanonicalPath": javax.management.AttributeNotFound	DMS Application(11.	App
Oct 21, 2009 6:38:33 AM PDT	Error		Failed to get "LocalAgentMonitored": javax.management.AttributeNo	DMS Application(11.	App
Oct 21, 2009 6:38:34 AM PDT	Error		Failed to get "MemberOf": javax.management.AttributeNotFoundExc	DMS Application(11.	App
Oct 21, 2009 6:38:34 AM PDT	Error		Failed to get "Parent": javax.management.AttributeNotFoundExcepti	DMS Application(11.	App
Oct 21, 2009 6:38:34 AM PDT	Error		Failed to get "HostName": javax.management.AttributeNotFoundExc	DMS Application(11.	App
Oct 21, 2009 6:41:25 AM PDT	Error		Failed to get "Members": javax.management.AttributeNotFoundExce	DMS Application(11.	App
Oct 21, 2009 8:09:45 AM PDT	Error		Failed to get "OracleHome": javax.management.AttributeNotFoundE	DMS Application(11.	App
Oct 21, 2009 8:09:45 AM PDT	Error		Failed to get "OracleInstance": javax.management.AttributeNotFoun	DMS Application(11.	App
Oct 21, 2009 8:09:45 AM PDT	Error		Failed to get "Version": javax.management.AttributeNotFoundExcept	DMS Application(11.	App
Oct 21, 2009 8:09:45 AM PDT	Error		Failed to get "DeleteCorrespondingJ2eeApp": javax.management.Att	DMS Application(11.	App
Oct 21, 2009 8:09:45 AM PDT	Error		Failed to get "DisplayName": javax.management.AttributeNotFoundE	DMS Application(11.	App

5. Trace the ECID to the earliest message. (You may need to increase the date or time range to view the first message with the ECID.)

12 Diagnosing Problems

This chapter describes how to use the Diagnostic Framework to collect and manage information about a problem so that you can resolve it or send it to Oracle Support.

This chapter contains the following topics:

- Understanding the Diagnostic Framework
- How the Diagnostic Framework Works
- Configuring the Diagnostic Framework
- Investigating, Reporting, and Solving a Problem

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:** Standardizing 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.
- **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 *Oracle Fusion Middleware 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 recorded in the ADR, but does not generate incident dumps. Flood-controlled incidents provide a way of informing you that a critical error is ongoing, without overloading the system with diagnostic data. You can choose to view or hide flood-controlled incidents when viewing incidents with ADRCI.

If more than 3 incidents with the same problem key occur within 15 minutes, subsequent incidents with the same problem key are flood controlled. That is, those subsequent incidents are created and written to the ADR, but no diagnostic data is captured.

12.1.2 Diagnostic Framework Components

The following topics describe that key components of the Diagnostic Framework:

- Automatic Diagnostic Repository
- Diagnostic Dumps
- Management MBeans
- ADCRI Command-Line Utility

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:

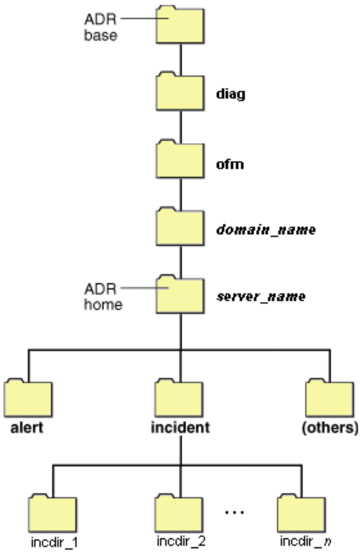
```
DOMAIN_HOME/servers/server_name/adr
```

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 an 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.

12.1.2.3 Management MBeans

The Diagnostic Framework provides MBeans that you can use to configure the Diagnostic Framework. For example, you can configure when flood control is enabled and 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 12.4.3.1, "Listing Diagnostic Dumps"
- Section 12.4.3.2, "Viewing a Description of a Diagnostic Dump"
- Section 12.4.3.3, "Executing Dumps"
- Section 12.4.4.1, "Creating an Incident Manually"
- "Diagnostic Framework Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

  To use the custom WLST Diagnostic Framework commands, you must invoke the WLST script from the Oracle Common home. See Section 3.5.1.1 for more information.

12.1.2.5 ADCRI 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) MW_HOME/wlserver_10.3/server/adr
(Windows) MW_HOME\wlserver_10.3\server\adr
```

See the following sections for information about using the ADCRI command-line utility:

- Packaging an Incident
- Purging Incidents

See Also:

- The chapter "ADRCI: ADR Command Interpreter" in the *Oracle Database Utilities*
- The chapter "Managing Diagnostic Data" in the *Oracle Database Administrator's Guide*

Both manuals are located at:

`http://www.oracle.com/technology/documentation/database.html`

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.

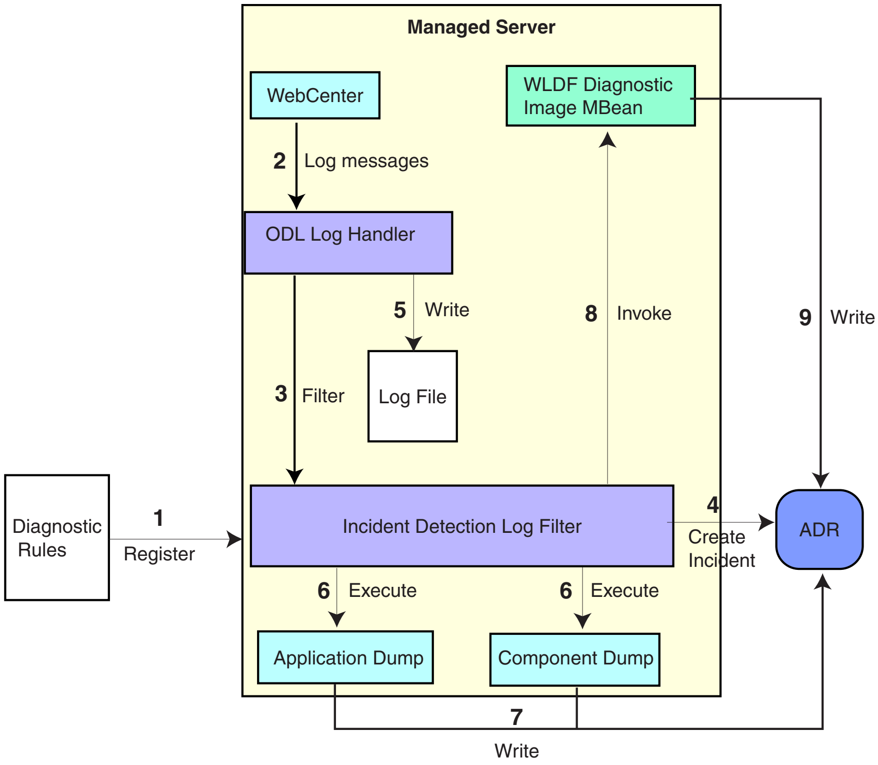
Incidents are automatically detected in two ways:

- 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 12.3.2.
- 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 (in this case Oracle WebCenter) 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 back to WebCenter.

   When an incident is created, a message, similar to the following, is written to the log file:

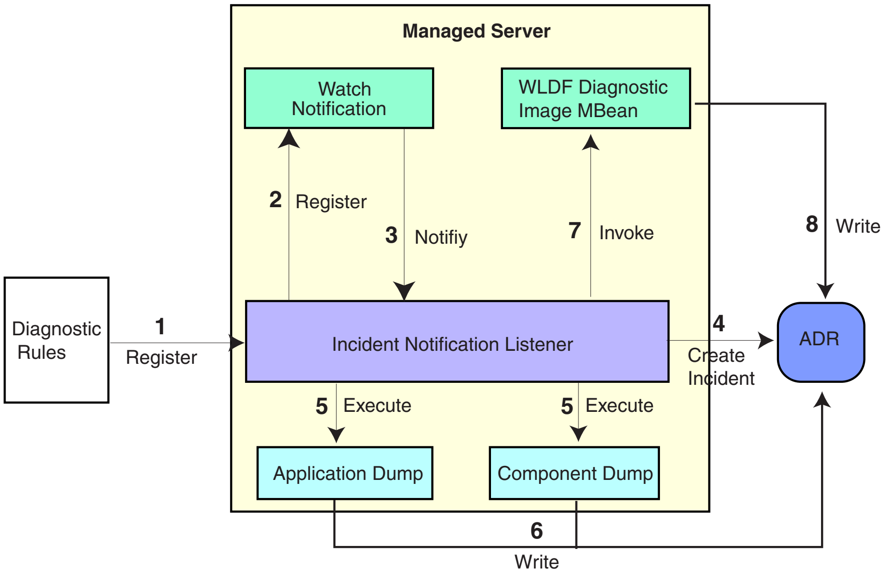
   ```
   [2009-09-16T06:37:59.264-07:00] [dfw] [NOTIFICATION] [DFW-40104] [oracle.dfw]
   [tid: 10] [ecid: 0000IF34gtMC8xT6uBf9EH1AgEck000000,0] [errid: 6]
   [detailLoc: /middleware/user_projects/base_
   domain/servers/AdminServer/adr/diag/ofm/base_domain/AdminServer]
   [probKey: MDS-123456 [testComponent][testModule]] incident 6 created with
    problem key "MDS-123456 [testComponent][testModule]", in directory
    /middleware/user_projects/base_domain/servers/AdminServer/adr/diag/ofm/base_
   domain/AdminServer/incident/incdir_6
   ```

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 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
- Enabling or disabling incident purging

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=DiagnosticsConfig
```

Table 12–1 shows the parameters for the DiagnosticConfig MBean and a description of each parameter.

Table 12–1 DiagnosticConfig MBean Parameters for Diagnostic Framework

Parameter	Description
incidentPurgeEnabled	Enables or disables the purging of incidents. Specify `true` for enabled or `false` for disabled. The default is `true`.
logDetectionEnabled	Enables or disables the detection of incidents through the log files. Specify `true` for enabled or `false` for disabled. The default is `true`.
floodControlEnabled	Enables or disables flood control. Specify `true` for enabled or `false` for disabled. The default is `true`. Note that flood control does not apply to manually created incidents.
floodControlIncidentCount	Sets the number of incidents with the same problem key that can occur within the time period, specified by floodControlIncidentTimeoutPeriod, before they are controlled by flood control. The default is 3. When flood control is enabled, if the number of incidents with the same problem key exceeds this count, an incident is created, but no diagnostics are captured with the incident.
floodControlIncidentTimeoutPeriod	Sets the time period in which the number of incidents, as specified by floodControlIncidentCount, with the same Problem Key can occur before they are controlled by flood control. The default is 15 minutes.

Table 12–1 (Cont.) DiagnosticConfig MBean Parameters for Diagnostic Framework

Parameter	Description
reservedMemoryKB	The amount of reserved memory that is released when OutOfMemoryError is detected. When the Diagnostic Framework starts, it allocates 512KB 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. The default is 512KB.
useExternalCommands	Indicates whether external JVM commands should be used to perform thread dumps. Specify `true` for enabled or `false` for disabled. The default is `true`.

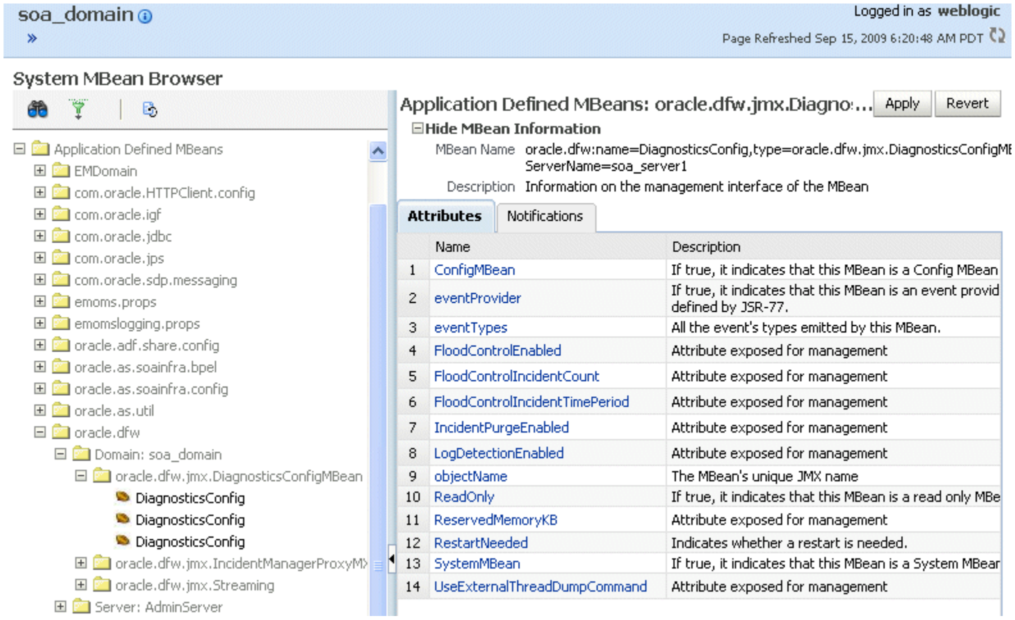
The following example shows how to configure these settings using the Fusion Middleware Control System MBean Browser:

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

   The System MBean Browser page is displayed.
4. Expand **Application Defined Beans**, then **oracle.dfw,** then **domain**.*domain_name,* then **dfw.jmx.DiagnosticsConfigMBean.**
5. Select one of the **DiagnosticConfig** entries. There is one DiagnosticConfig entry for each server.

   In the Application Defined MBean pane, expand **Show MBean Information** to see the server name.

   The following shows the System MBean Browser page:



MBean Name oracle.dfw:name=DiagnosticsConfig,type=oracle.dfw.jmx.DiagnosticsConfigMB ServerName=soa_server1

Description Information on the management interface of the MBean

	Name	Description
1	ConfigMBean	If true, it indicates that this MBean is a Config MBean
2	eventProvider	If true, it indicates that this MBean is an event provid defined by JSR-77.
3	eventTypes	All the event's types emitted by this MBean.
4	FloodControlEnabled	Attribute exposed for management
5	FloodControlIncidentCount	Attribute exposed for management
6	FloodControlIncidentTimePeriod	Attribute exposed for management
7	IncidentPurgeEnabled	Attribute exposed for management
8	LogDetectionEnabled	Attribute exposed for management
9	objectName	The MBean's unique JMX name
10	ReadOnly	If true, it indicates that this MBean is a read only MBe
11	ReservedMemoryKB	Attribute exposed for management
12	RestartNeeded	Indicates whether a restart is needed.
13	SystemMBean	If true, it indicates that this MBean is a System MBear
14	UseExternalThreadDumpCommand	Attribute exposed for management

6. To change the values for the attributes listed in Table 12–1, select the attribute.

7. Enter or select the value in the **Value** field.
8. Click **Apply.**

12.3.2 Configuring WLDF Watch and Notification for the Diagnostic Framework

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:

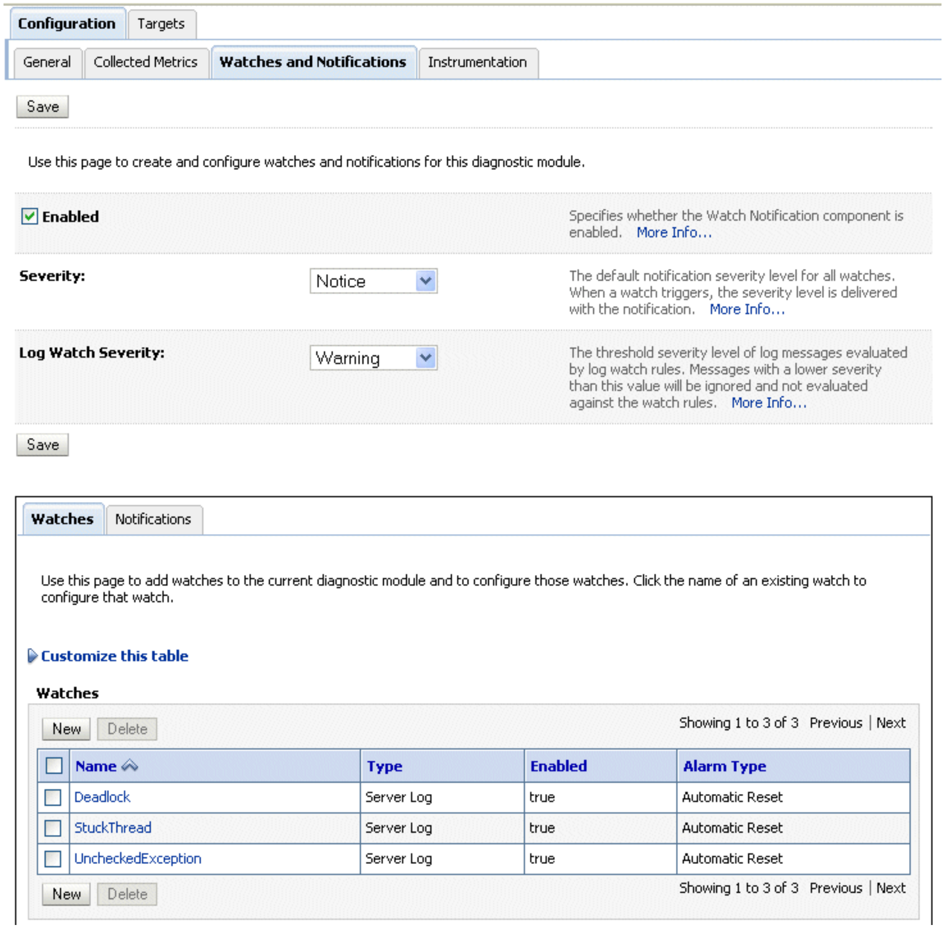
Name	Description
Deadlock	Two or more Java threads have circular lock chains among their Java Monitor object usage.
StuckThread	An Oracle WebLogic Server ExecuteThread, which is blocked or busy for more than the time specified by the Oracle WebLogic Server StuckThreadMaxTime parameter.
UncheckedException	This category includes all Unchecked Exception, RuntimeException, and Errors caught by the Oracle WebLogic Server ExecuteThread, such as NullPointerException, StackOverflowError, or OutOfMemoryError.

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 in order to create an incident:

1. Display the Administration Console, as described in Section 3.4.1.
2. In the Change Center, click **Lock & Edit.**
3. In the left pane, expand **Diagnostics** and select **Diagnostic Modules.**

   The Summary of Diagnostic Modules page is displayed.
4. Click **Module-FMWDFW.**

   The Settings for Module-FMWDFW page is displayed.
5. Select the Watches and Notifications tab, which is shown in the following figure:



Configuration | 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 3 of 3 Previous | Next

□	Name	Type	Enabled	Alarm Type
□	Deadlock	Server Log	true	Automatic Reset
□	StuckThread	Server Log	true	Automatic Reset
□	UncheckedException	Server Log	true	Automatic Reset

New | Delete — Showing 1 to 3 of 3 Previous | Next

6. Select the Watches tab and click **New.**

   The Create Watch page is displayed.

7. For **Name,** enter a name for the watch.
8. For **Watch Type,** select a type.
9. Click **Next.**
10. For **Current Watch Rule,** construct an expression. For example, (SEVERITY = 'Error') AND (MSGID = 'BEA-000337').
11. Click **Next.**
12. Select an alarm type.
13. For **Notifications,** select **FMWDFW-notification.**
14. Click **Finish.**

For more information on creating watches, see "Construct watch rule expressions" in the Administration Console Online Help.

12.4 Investigating, Reporting, and Solving a Problem

This section describes how to use WLST and ADRCI commands 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:

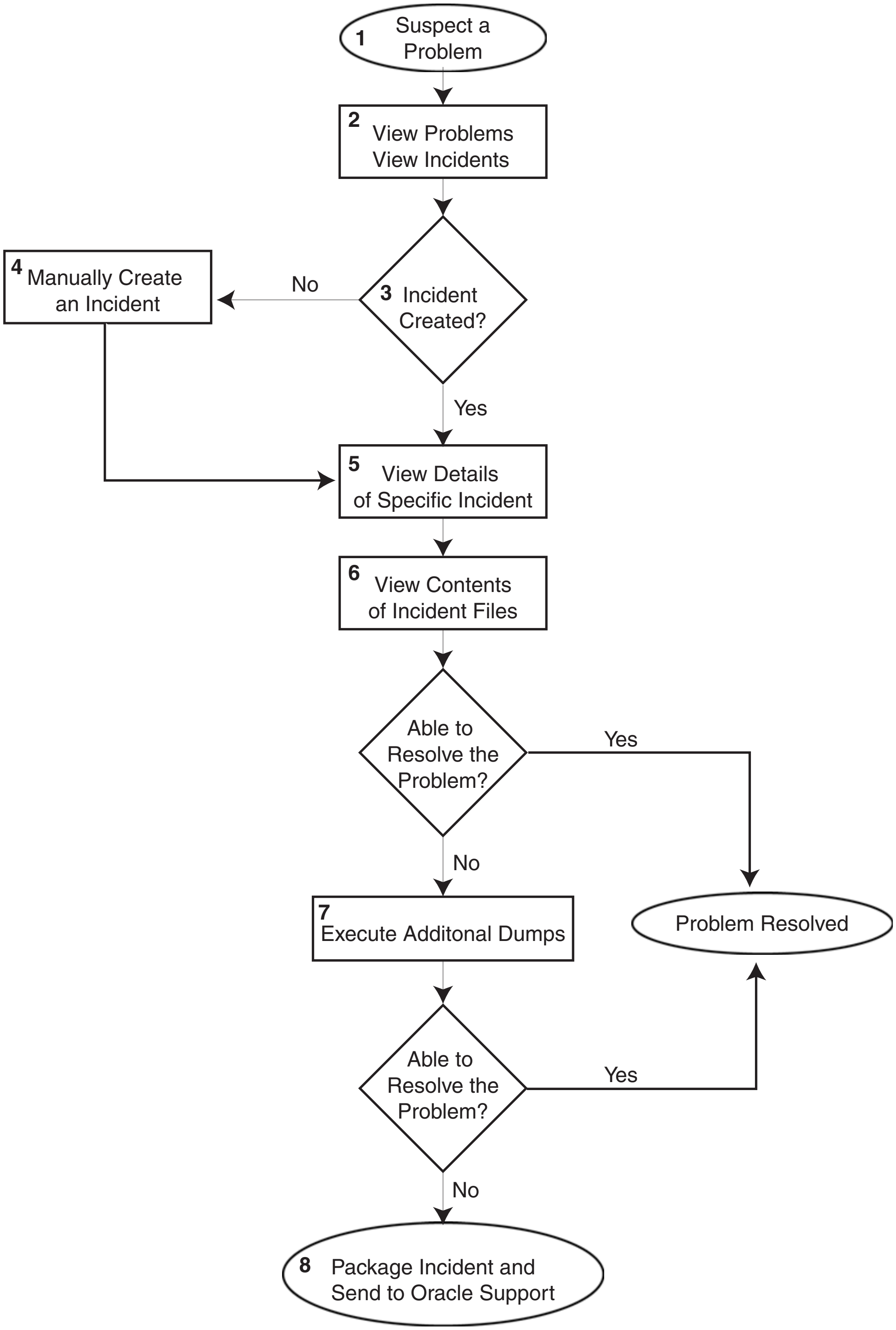
- Roadmap—Investigating, Reporting, and Resolving a Problem
- Viewing Problems and Incidents
- Working with Diagnostic Dumps
- Managing Incidents

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 in some cases, resolve a problem.

Figure 12–4 Flow for Investigating a Problem


1 Suspect a Problem
2 View Problems View Incidents
3 Incident Created?
No
4 Manually Create an Incident
Yes
5 View Details of Specific Incident
6 View Contents of Incident Files
Able to Resolve the Problem?
Yes
No
7 Execute Additonal Dumps
Problem Resolved
Able to Resolve the Problem?
Yes
No
8 Package Incident and Send to Oracle Support

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 12.4.4.1 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 12.4.3 for information about the dumps available and how to execute them.

8. If you still cannot resolve the problem, package the incident and send it to Oracle Support. See Section 12.4.4.2 for more information.

12.4.2 Viewing Problems and Incidents

You can view the set of problems, the list of incidents and view the details of a particular incident using the WLST command-line utility, as described in the following topics:

- Viewing Problems
- Viewing Incidents

12.4.2.1 Viewing Problems

You can view the set of problems by using 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 view the list of incidents and view the details of a particular incident using the WLST command-line utility.

To view the list of all available incidents or the incidents related to a specific problem, use the WLST `listIncidents` command, using the following format:

```
listIncidents([id], [ADRHome])
```

For example, to see the list of all incidents, use the following command:

```
listIncidents()
Incident Id     Problem Key             Incident Time
        2       BEA-101020 [HTTP]       Fri Sep 18 13:42:01 PDT 2009
        1       BEA-101020 [HTTP]       Tue Sep 15 06:17:39 PDT 2009
```

To view the incidents related to a specific problem, use the following command:

```
listIncidents(id='1')
Incident Id     Problem Key             Incident Time
        2       BEA-101020 [HTTP]       Fri Sep 18 13:42:01 PDT 2009
        1       BEA-101020 [HTTP]       Tue Sep 15 06:17:39 PDT 2009
```

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: BEA-101020 [HTTP]
Incident Time: Tue Sep 15 06:17:39 PDT 2009
Error Message Id:  BEA-101020
Execution Context: 0000IExqUvyAhKB5JZ4Eyf1Afdj600009i
Flood Controlled: false
Dump Files :
    dms_ecidctx1_i1.dmp
    jvm_threads2_i1.dmp
    dms_metrics3_i1.dmp
    odl_logs4_i1.dmp
    odl_logs5_i1.dmp
    diagnostic_image_AdminServer_2009_09_15_06_17_42.zip
    readme.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_logs4_i1.dmp', outputFile='/tmp/odl_logs4_i1_
dmp.output')
```

The output is written to the file odl_logs4_i1_dmp.output.

12.4.3 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

12.4.3.1 Listing Diagnostic Dumps

You can find a list of diagnostic dumps that are available for a Managed Server by specifying the WLST `listDumps` command, using the following format:

```
listDumps([appName] [,server])
```

For example, to list the available dumps for soa_server1:

```
listDumps(server='soa_server1')
Location changed to domainRuntime tree. This is a read-only tree with DomainMBean
as the root.
For more help, use help(domainRuntime)

jvm.classhistogram
dms.ecidctx
wls.image
odl.logs
dms.metrics
jvm.threads

Use the command describeDump(name=<dumpName>) for help on a specific dump.
```

Table 12–2 lists the diagnostic dump actions that are defined by Oracle Fusion Middleware and their descriptions.

Table 12–2 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 "DMS Internal Metrics" in the *Oracle Fusion Middleware Performance Guide.*
jvm.classhistogram	A JVM class histogram, the output of which varies depending on the JVM vendor.
jvm.threads	Summary statistics about the threads running in a JVM as well as performing a full thread dump.
odl.logs	Contents of diagnostic logs, correlated by ECID or time range.

Table 12–2 (Cont.) Diagnostic Dump Actions

Dump Action	Description
wls.image	The WLDF server image dump.

12.4.3.2 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.
Mandatory Arguments:
Optional Arguments:
    Name        Type     Description
    format      STRING   Format of the dump output; raw or xml
wls:/soa_domain/serverConfig>
```

12.4.3.3 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 12.4.3.2.

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_i1.dmp added to incident 1
```

The dump output is written to the information about incident 1. If you execute the showIncident command for incident 1, the output includes dms_metrics1_i1.dmp.

12.4.4 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
- Packaging an Incident
- Purging Incidents

12.4.4.1 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.

The following describes how to manually create an incident based on a message ID:

1. Search the log files, as described in Section 11.3.1. 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('weblogic', '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: 55
   Problem Id: 4
   Problem Key: MDS-50500 [MANUAL]
   Incident Time: 25th June 2009 11:55:45 GMT
   Error Message Id: MDS-50500
   Flood Controlled: false
   ```

   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='soa_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 12.4.3.

12.4.4.2 Packaging an Incident

You can package the incident to facilitate sending the information to Oracle Support by using the ADR Command Interpreter (ADRCI). 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) MW_HOME/wlserver_10.3/server/adr
(Windows) MW_HOME\wlserver_10.3\server\adr
```

Packaging an incidents 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 to point to the following directory:

      ```
      MW_HOME/wlserver_10.3/server/adr
      ```

2. Invoke ADRCI. For example:

   ```
   MW_HOME/wlserver_10.3/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/oracle1/Oracle/Middleware/user_projects/domains/soa_
   domain/servers/soa_server1/adr
   SET HOMEPATH diag/ofm/soa_domain/soa_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_20090925132315_COM_1.zip, mode
   complete
   ```

   This generates a complete physical package (zip file) in the designated path.

   See Also: The "ADRCI: ADR Command Interpreter" chapter of the *Oracle Database Utilities*, which is located at:

   http://www.oracle.com/technology/documentation/database.html

12.4.4.3 Purging Incidents

By default, incidents are automatically purged (deleted) if they are older than 30 days.

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*, which is located at:

```
http://www.oracle.com/technology/documentation/database.html
```

You can set the purge frequency, as described in Section 12.3.

Part VI

Advanced Administration

This part describes advanced administration tasks that involve reconfiguring Oracle Fusion Middleware.

It contains the following chapters:

- Chapter 13, "Managing the Metadata Repository"
- Chapter 14, "Changing Network Configurations"

13 Managing the Metadata Repository

This chapter provides information on managing the metadata repositories used by Oracle Fusion Middleware.

It contains the following topics:

- Understanding a Metadata Repository
- Creating a Database-Based Metadata Repository
- Managing the MDS Repository
- Managing Metadata Repository Schemas

13.1 Understanding a Metadata Repository

A metadata repository contains metadata for Oracle Fusion Middleware components, such as Oracle BPEL Process Manager, Oracle B2B, and Oracle WebCenter. 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 BPEL Process Manager or Oracle Internet Directory.)

A particular type of repository, the Oracle Metadata Service (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, such as Oracle B2B. For information related specifically to the MDS Repository type, see Section 13.3.

You can create a database-based repository or, for MDS, a file-based repository. For production environments, you use a database-based repository. Most components, such as Oracle BPEL Process Manager, require that a schema be installed in a database, necessitating the use of a database-based repository.

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 multiple repositories in a single database. You can use it to create the MDS Repository or a repository for metadata for particular components, such as Oracle WebCenter. RCU creates the necessary schemas for the components. See Appendix D for a list of the schemas and their tablespaces and datafiles.

With RCU, you can also drop component schemas.

For information about the supported versions of database platforms and versions, and other prerequisites for the database, see:

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

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 a database-based MDS Repository, see Section 13.3.

See Also: *Oracle Fusion Middleware Repository Creation Utility User's Guide* for information about how to use RCU to create a database-based metadata repository

13.3 Managing the MDS Repository

Oracle Metadata Service (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 B2B and 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.
- You have a custom Oracle WebCenter application.
- You have a SOA composite application (SCA).

The following topics provide information about the MDS Repository:

- Understanding the MDS Repository
- Registering and Deregistering a Database-Based Metadata Repository
- Registering and Deregistering a File-Based Metadata Repository
- Viewing Information about an MDS Repository
- Listing Repositories and Partitions
- Configuring an Application to Use a Different MDS Repository or Partition
- Moving Metadata from a Test System to a Production 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: *Oracle Fusion Middleware 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 the *Oracle Fusion Middleware Fusion Developer's Guide for 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 essentially a set of criteria that restrict the search results to certain set of attribute types and values, component types, text content, and metadata paths. The API allows multiple conditions to be combined together 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: The 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.
- The capability in a running environment to isolate metadata changes 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 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 in Managed Server A, the data is written to the database-based repository. The application deployed in Managed Server B uses the updated customizations. This supports high availability (in particular, active/active scenarios.)

- Clustered updates: Updates from multiple hosts to the metadata are allowed. 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.

The MDS Repository supports Oracle databases, as well as non-Oracle databases. For more information about the supported databases, see:

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

Note: To create a metadata repository in SQL Server, set READ_COMMITTED_SNAPSHOT to ON for the hosting database. This enables the needed row versioning support. This feature can be enabled using SQL command ALTER DATABASE, as in the following example:

```
ALTER DATABASE mds SET READ_COMMITTED_SNAPSHOT ON
```

Furthermore, you should use case sensitive collation to support the case-sensitive semantics in the metadata repository. For example, if Latin1_General is used, select the SQL_Latin1_General_CP1_CS_AS collation using the following SQL command:

```
ALTER DATABASE mds COLLATE SQL_Latin1_General_CP1_CS_AS
```

In the example, the substring of "CS" denotes case sensitivity.

Also note that when installed on SQL Server, Oracle Content Server 10g requires a database instance with case insensitive collation. As a result, it cannot exist in the same SQL Server database instance as MDS. If you wish to install both MDS and Oracle Content Server 10g, you must create a separate database instance with case insensitive collation for installing the Oracle Content Server 10g schema.

Note that there are some minor differences between an Oracle schema and a SQL Server schema. The length of the certain text fields are shorter for a SQL Server schema. For example, the full path name of the metadata in SQL Server is limited to 400 characters.

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.

13.3.1.1 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 following sections 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 following sections:

- 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. The directory or archive must be accessible from the host where the application is running. 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. The directory or archive must be accessible from the host where the application is running. 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 3.5.1.1 for more information.
> - For more information about the WLST commands used in these sections, see "Metadata Services (MDS) Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

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
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

Table 13–1 (Cont.) MDS Operations and Required Roles

Operation	Logical Role
Deregister metadata file repository	Admin role for domain
Export metadata	Operator role for application
Import MAR	Admin role for application
Import metadata	Admin role for application
List metadata label	Monitor role for application
Promote metadata label	Admin role for application
Purge metadata	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 the *Oracle Fusion Middleware Security Guide*.

13.3.2 Registering and Deregistering a Database-Based Metadata Repository

The following sections describe how to register and deregister a database-based MDS Repository:

- Registering a Database-Based MDS Repository
- Deregistering a Database-Based MDS Repository

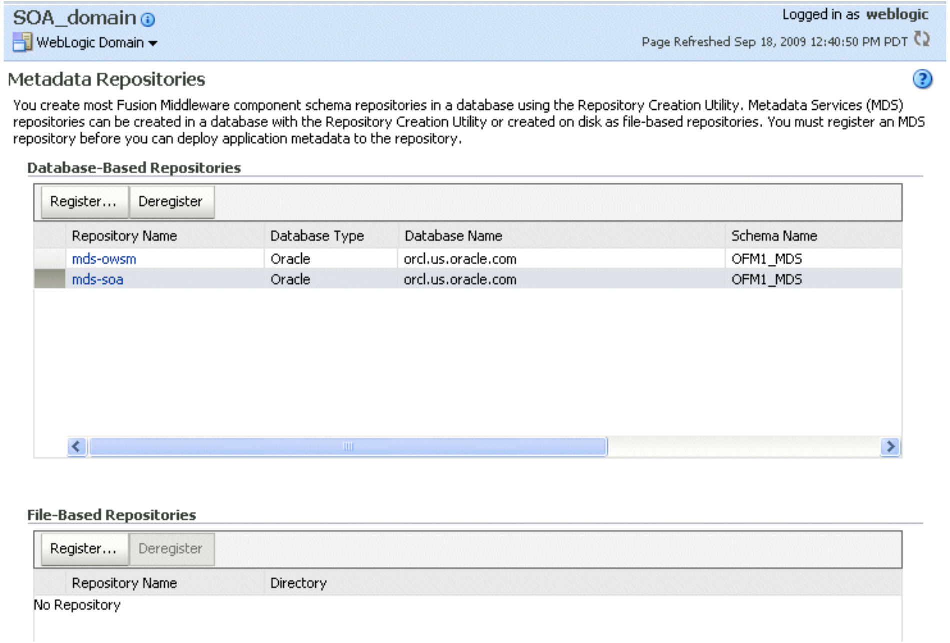
13.3.2.1 Registering a Database-Based MDS Repository

You create a database-based MDS Repository using RCU, as described in Section 13.2. Before you can deploy an application to an MDS Repository, you must register the repository with the Oracle WebLogic Server domain.

To register a database-based MDS Repository using Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **WebLogic Domain.**
2. Select the domain.
3. From the WebLogic Domain menu, choose **Metadata Repositories.**

   The Metadata Repositories page is displayed, as shown in the following figure:



Metadata Repositories

You create most Fusion Middleware component schema repositories in a database using the Repository Creation Utility. Metadata Services (MDS) repositories can be created in a database with the Repository Creation Utility or created on disk as file-based repositories. You must register an MDS repository before you can deploy application metadata to the repository.

Database-Based Repositories

Repository Name	Database Type	Database Name	Schema Name
mds-owsm	Oracle	orcl.us.oracle.com	OFM1_MDS
mds-soa	Oracle	orcl.us.oracle.com	OFM1_MDS

File-Based Repositories

Repository Name	Directory
No Repository	

4. In the Database-Based Repositories section, click **Register.**

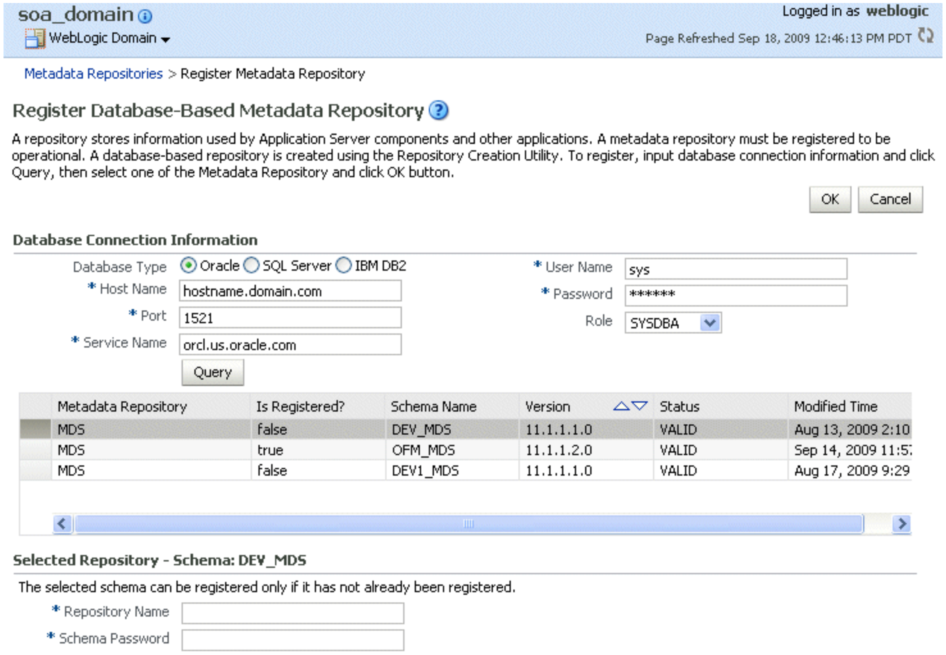
   The Register Database-Based Metadata Repository page is displayed.

5. In the Database Connection section, enter the following information:

   - For **Database,** 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.**

6. Click **Query.**

   A table is displayed that lists the schemas and their metadata repositories in the database, as shown in the following figure:


soa_domain
WebLogic Domain
Logged in as weblogic
Page Refreshed Sep 18, 2009 12:46:13 PM PDT
Metadata Repositories > Register Metadata Repository
OK
Cancel
Query

Register Database-Based Metadata Repository

A repository stores information used by Application Server components and other applications. A metadata repository must be registered to be operational. A database-based repository is created using the Repository Creation Utility. To register, input database connection information and click Query, then select one of the Metadata Repository and click OK button.

Database Connection Information

Database Type	● Oracle ○ SQL Server ○ IBM DB2	* User Name	sys
* Host Name	hostname.domain.com	* Password	******
* Port	1521	Role	SYSDBA
* Service Name	orcl.us.oracle.com		

Metadata Repository	Is Registered?	Schema Name	Version	Status	Modified Time
MDS	false	DEV_MDS	11.1.1.1.0	VALID	Aug 13, 2009 2:10
MDS	true	OFM_MDS	11.1.1.2.0	VALID	Sep 14, 2009 11:5
MDS	false	DEV1_MDS	11.1.1.1.0	VALID	Aug 17, 2009 9:29

Selected Repository - Schema: DEV_MDS

The selected schema can be registered only if it has not already been registered.

* Repository Name	
* Schema Password	

7. 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.

8. Click **OK.**

   The repository is registered with the Oracle WebLogic Server domain. In addition, a system data source is created with the name mds-*repository_name*. Global transaction support is disabled for the data source.

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 Deregistering a Database-Based MDS Repository

Deregistration does not result in loss of data stored in the repository. However, any applications using this repository would not function after the repository is deregistered. You must ensure that no application is using the repository before you deregister it.

To deregister an MDS Repository using Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **WebLogic Domain.**
2. Select the domain.
3. From the WebLogic Domain menu, choose **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.

4. Select the repository from the table.
5. Click **Deregister.**
6. Click **Yes** in the Confirmation dialog box.

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 Metadata Repository

The following topics describe how to register and deregister a file-based metadata repository:

- Creating and Registering a File-Based Metadata Repository
- Deregistering a File-Based Repository

13.3.3.1 Creating and Registering a File-Based Metadata Repository

You can create a file-based MDS Repository and register it with an Oracle WebLogic Server domain using Fusion Middleware Control.

To register a file-based repository using Fusion Middleware Control:

1. From the navigation pane, expand the farm, then **WebLogic Domain.**
2. Select the domain.
3. From the WebLogic Domain menu, choose **Metadata Repositories.**

   The Metadata Repositories page is displayed.
4. In the File-Based Repository section, click **Register.**

   The Register Metadata Repository page is displayed.
5. 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. The directory is created if it does not exist.

     Note the following:

     – If the path specified 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 or the path exists as a file not a directory, an error is displayed and the registration fails.

6. Click **OK.**

The repository is created and registered. It is now 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 3.6.

13.3.3.2 Deregistering a File-Based 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 navigation pane, expand the farm, then **WebLogic Domain.**
2. Select the domain.
3. From the WebLogic Domain menu, choose **Metadata Repositories.**

   The Metadata Repositories page is displayed.
4. In the File-Based Repository section, select the repository and click **Deregister.**
5. 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.

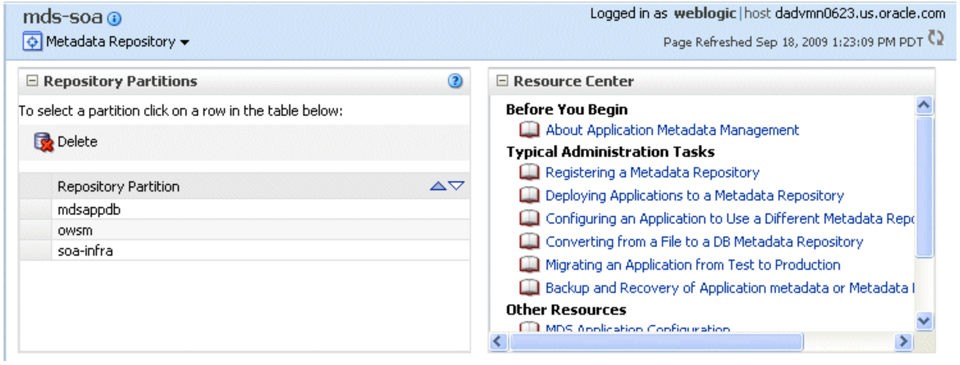
You can also deregister a file-based repository using system MBeans. For information about using the System MBean Browser, see Section 3.6.

13.3.4 Viewing Information about an MDS Repository

To view information about an MDS Repository with Fusion Middleware Control:

1. From the navigation pane, expand the farm and then expand **Metadata Repositories.**
2. Select the repository.

   The following figure shows the home page for an MDS Repository:



mds-soa

Metadata Repository

Logged in as **weblogic** | host dadvmn0623.us.oracle.com

Page Refreshed Sep 18, 2009 1:23:09 PM PDT

Repository Partitions

To select a partition click on a row in the table below:

Delete

Repository Partition
mdsappdb
owsm
soa-infra

Resource Center

Before You Begin

About Application Metadata Management

Typical Administration Tasks

Registering a Metadata Repository

Deploying Applications to a Metadata Repository

Configuring an Application to Use a Different Metadata Repo

Converting from a File to a DB Metadata Repository

Migrating an Application from Test to Production

Backup and Recovery of Application metadata or Metadata I

Other Resources

MDS Application Configuration

3. To see general information about the repository, such as the type of repository and location, click the **Info** icon in the context pane, as described in Section 3.3.3.

13.3.5 Listing Repositories and Partitions

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. In Fusion Middleware Control, from the navigation pane, navigate to the domain and select it. 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 to view the information.

13.3.6 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 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 13.3.6.1 describes how to clone a partition and how to redeploy the application, specifying the partition that you have cloned.

  - Export the metadata from the current partition and import the metadata into the new partition, then redeploy the application, specifying a new partition.

    Section 13.3.6.2 describes how to redeploy the application, specifying a 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 13.3.6.3 describes how to change the system data source.

13.3.6.1 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:

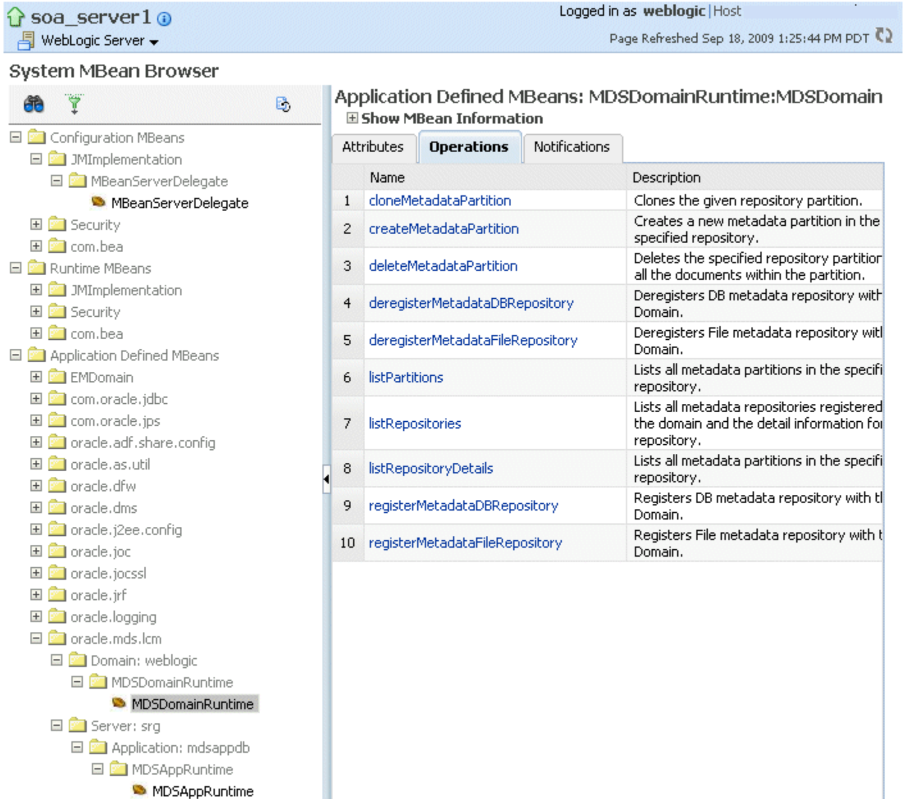
a. In Fusion Middleware Control, from the navigation pane, navigate to the Managed Server from which the application is deployed. From the WebLogic Server menu, choose **System MBean Browser.**

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:



	Name	Description
1	cloneMetadataPartition	Clones the given repository partition.
2	createMetadataPartition	Creates a new metadata partition in the specified repository.
3	deleteMetadataPartition	Deletes the specified repository partitior all the documents within the partition.
4	deregisterMetadataDBRepository	Deregisters DB metadata repository witl Domain.
5	deregisterMetadataFileRepository	Deregisters File metadata repository witl Domain.
6	listPartitions	Lists all metadata partitions in the specifi repository.
7	listRepositories	Lists all metadata repositories registered the domain and the detail information foi repository.
8	listRepositoryDetails	Lists all metadata partitions in the specifi repository.
9	registerMetadataDBRepository	Registers DB metadata repository with tl Domain.
10	registerMetadataFileRepository	Registers File metadata repository with t Domain.

d. Select **cloneMetadataPartiton.**

The Operation: cloneMetadataPartiton 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, Section 9.5.3, or Section 9.6.3 depending on the type of application. 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.**

13.3.6.2 Creating a New Partition and Reassociating the Application to It

You can create a new partition in the new 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, Section 9.5.3, or Section 9.6.3 depending on the type of application. 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/mypartiton', docs='/**')
```

4. Optionally, deregister the original repository, as described in Section 13.3.3.2 or Section 13.3.2.2.

Alternatively, you can create 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')
```

13.3.6.3 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 use Oracle WebLogic Server Administration Console:

1. In the Change Center, click **Lock & Edit.**
2. In the Domain Structure section, expand **Services,** then **JDBC**, and select **Data Sources.**

   The Summary of JDBC Data Sources page is displayed.
3. Select the data source you want to change.

   The Settings page is displayed.
4. Select the Connection Pool tab.
5. To change the database, modify the **URL** field. For example:

   ```
   jdbc:oracle:thin:@hostname.domainname.com:1522/orcl
   ```
6. To change the schema, modify the Properties field, changing the `user` property. For example, to specify a schema named OFM-MDS, use the following:

   ```
   user=OFM-MDS
   ```
7. For **Password,** enter the password for the schema, then confirm the password.
8. Click **Save.**
9. Restart the servers that use this data source. (Click the Target tab to see the servers that use this data source.)

13.3.7 Moving Metadata from a Test System to a Production System

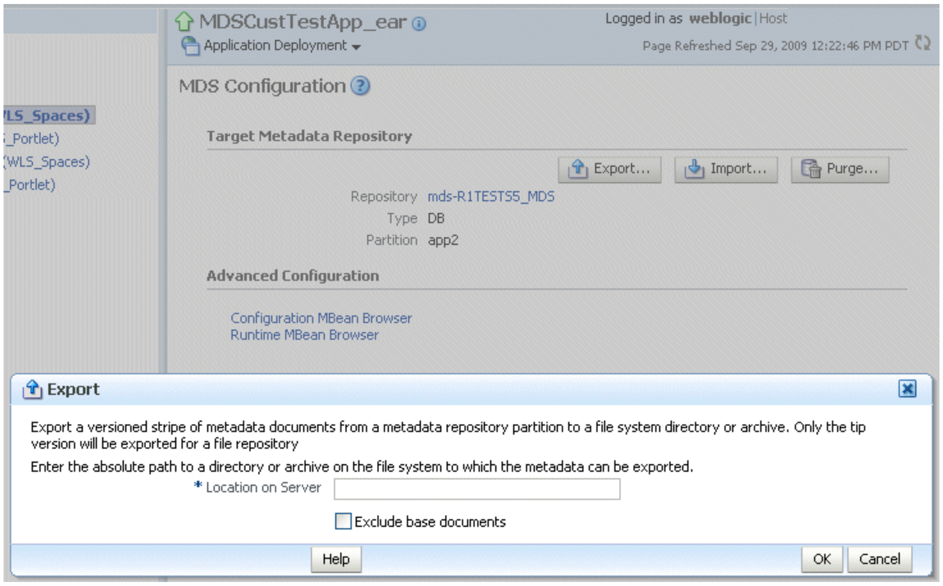
You can transfer the metadata in MDS from one partition to another. As an example, you want to move an application from a test system to a production system. You have a test application that is deployed in a domain in the test system and a production application deployed in a domain in the production 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.

To use Fusion Middleware Control to transfer metadata:

1. From the navigation pane, expand the farm, expand **Application Deployments**, then select the application.
2. From the Application Deployment menu, choose **MDS Configuration.**

   The MDS Configuration page is displayed.
3. Click **Export.** The Export dialog box is displayed, as shown in the following figure:



Repository mds-R1TESTS5_MDS
Type DB
Partition app2

Export a versioned stripe of metadata documents from a metadata repository partition to a file system directory or archive. Only the tip version will be exported for a file repository

Enter the absolute path to a directory or archive on the file system to which the metadata can be exported.

* Location on Server

□ Exclude base documents

4. In the Export dialog box, enter a directory location or archive to which the metadata can be exported. The directory or archive file must be a local or network directory or file where the application is physically deployed. If the location does not exist in the file system, a directory is created except that, when the name ends with .jar, .JAR, .zip, or .ZIP, an archive is created. Exporting metadata to an existing archive overwrites the existing file.

   If you specify a directory, the metadata is written to a subdirectory of the directory that you specified, with the name of the partition that was exported as the name of the subdirectory.

   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.

   Click **OK.**

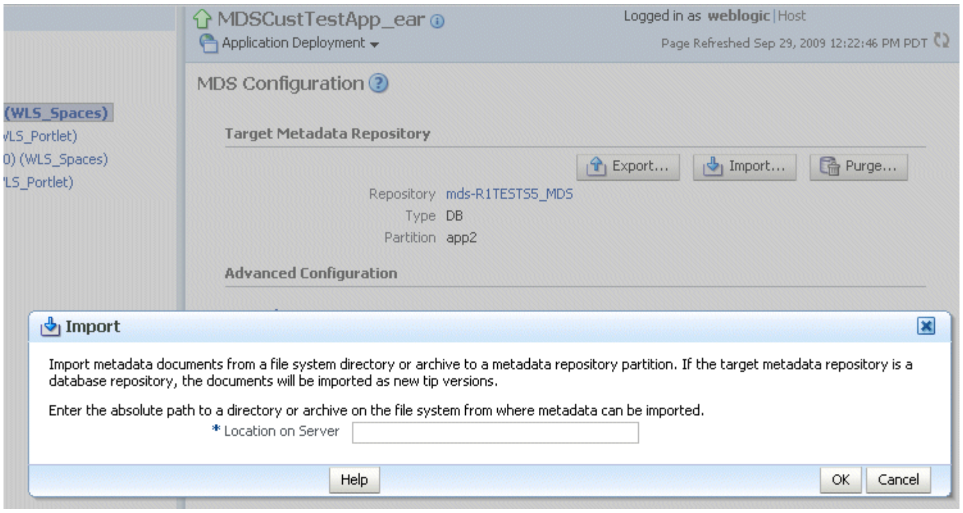
5. In the Confirmation dialog box, click **Close.**

6. If the production application is on a different system, copy the exported metadata to that system.

7. From the navigation pane for the production system, expand the farm, expand **Application Deployments**, then select the application.

8. Click **Import.**

   The Import dialog box is displayed, as shown in the following figure:



Repository mds-R1TESTS5_MDS
Type DB
Partition app2

Import metadata documents from a file system directory or archive to a metadata repository partition. If the target metadata repository is a database repository, the documents will be imported as new tip versions.

Enter the absolute path to a directory or archive on the file system from where metadata can be imported.

* Location on Server

9. In the Import dialog box, 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.
10. Click **OK.**

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.
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\`.

13.3.8 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, Section 9.5.3, or Section 9.6.3 depending on the type of application. 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.

13.3.9 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.

13.3.9.1 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 repository1, use the following command:

```
deleteMetadataPartition(repository='mds-repos1', partition='partition1')
```

13.3.9.2 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 the farm and then 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 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 labeled documents, you must first delete the label, as described in Section 13.3.11.2.

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.

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:

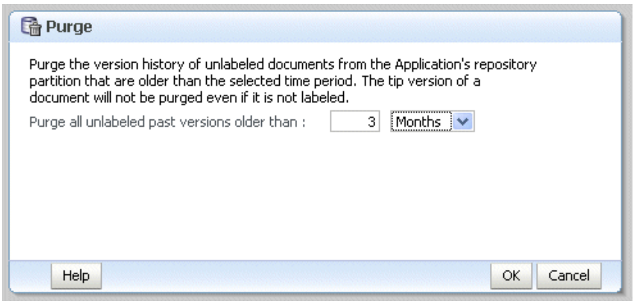
```
purgeMetadata(application='sampleApp', server='server1', olderThan=100)
```

To use Fusion Middleware Control to purge the metadata version history:

1. From the navigation pane, expand the farm, expand **Application Deployments**, then select the application.
2. From the Application Deployment menu, choose **MDS Configuration.**

   The MDS Configuration page is displayed.
3. Click **Purge.**

   The Purge dialog box is displayed, as shown in the following figure:



Purge

Purge the version history of unlabeled documents from the Application's repository partition that are older than the selected time period. The tip version of a document will not be purged even if it is not labeled.

Purge all unlabeled past versions older than : 3 Months

4. In the **Purge all unlabeled past versions that are older than** field, enter a number and select the unit of time. For example, enter **3** and select **months.**
5. Click **OK.**

   A progress box is displayed. When the operation completes, a completion box is displayed.
6. Click **Close.**

13.3.11 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.

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.

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 commands to support logical backup and recovery of an application's metadata contained in the partition.

Labels are supported only in database-based repositories.

The following topics describe how to manage labels:

- Creating Metadata Labels
- Deleting Metadata Labels
- Listing Metadata Labels
- Promoting Metadata Labels

13.3.11.1 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')
```

If the application has more than one version, you must use the applicationVersion parameter to specify the version.

13.3.11.2 Deleting Metadata Labels

To delete a 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 13.3.11.3.

13.3.11.3 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')
```

If the application has more than one version, you must use the `applicationVersion` parameter to specify the version.

13.3.11.4 Promoting Metadata Labels

You can promote documents associated with a metadata label so that they are now the last 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')
```

If the application has more than one version, you must use the `applicationVersion` parameter to specify the version.

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 **JDBC,** and select **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/technology/documentation/database.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 will ultimately save you time and money as well as give 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

14

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.

This chapter includes the following topics:

- Changing the Network Configuration
- 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

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. When you change the host name, domain name, or IP address of Oracle WebLogic Server, you also automatically change the information for Java components, such as Oracle SOA Suite and Oracle WebCenter components that are deployed to Oracle WebLogic Server.
- Oracle Fusion Middleware Web Tier components, Oracle Web Cache and Oracle HTTP Server. You can change the host name or the IP address.
- Oracle Database

The following topics describe how to change the host name, domain name, or IP address:

- Changing the Network Configuration of a Managed Server
- Changing the Network Configuration of Web Tier Components

14.1.1 Changing the Network Configuration of a Managed Server

To change the host name, domain name, or IP address of a Managed Server:

1. Display the Administration Console, as described in Section 3.4.1.
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 the Help.

4. 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.**

5. 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.

14.1.2 Changing the Network Configuration of Web Tier Components

If you change the host name, domain name, or IP address of a host that contains multiple Oracle instances, you must change the network configuration of each Oracle instance that resides on that host. You do not need to make changes to any system component that resides on another host.

You can change the network configuration of Oracle HTTP Server and Oracle Web Cache by using the following command:

```
(UNIX) ORACLE_HOME/chgip/scripts/chpiphost.sh
(Windows) ORACLE_HOME\chgip\scripts\chpiphost.bat
```

The format of the command is:

```
chgiphost.sh | chgiphost.bat
             [-noconfig] [-version] [-help]
             [ -oldhost old_host_name -newhost new_host_name]
             [-oldip old_IP_address -newip new_IP_address]
              -instanceHome Instance_path
```

The parameters have the following meanings:

- `noconfig`: The default for changing the network parameters.
- `version`: Displays the version of the chgiphost tool.
- `help`: Displays help for the command.
- `oldhost`: The fully qualified name of the old host. Use this parameter, with `newhost`, to change the host name or domain name, or both.
- `newhost`: The fully qualified name of the new host. Use this parameter, with `oldhost`, to change the host name or domain name, or both.
- `oldip`: The old IP address.
- `newip`: The new IP address.
- `instanceHome`: The full path of the Oracle instance.

For example, to change the host name, domain name, and IP address of a host that contains either Oracle HTTP Server or Oracle Web Cache, or both, take the following steps:

Task 1 Prepare Your Host

Prepare your host for the change:

1. Perform a backup of your environment before you start this procedure. See Chapter 16.
2. Shutdown all Oracle Fusion Middleware processes. See Chapter 4.

Task 2 Change the Host Name, Domain Name, or IP Address

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.

1. Make the updates to your operating system to properly change the host name, domain name, or IP address.
2. Restart the host, if necessary for 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 to ensure that everything is resolving properly.

Task 3 Run the chgiphost Command

Follow these steps for each Oracle instance that contains Oracle HTTP Server or Oracle Web Cache on your host. Be sure to complete the steps entirely for one Oracle instance before you move on to the next.

1. Log in to the host as the user that installed Oracle Fusion Middleware.
2. Run the chgiphost command.

   The following example changes the host name from host_a to host_b and the domain name from dom_1 to dom_2 for an Oracle instance named inst_a. It also changes the IP address:

   ```
   chgiphost.sh  -noconfig
                -oldhost host_a.dom_1 -newhost host_b.dom_2
                -instanceHome /scratch/Oracle/Middleware/inst_a
   ```

Task 4 Restart Processes

Restart all Oracle Fusion Middleware processes. See Chapter 4.

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 components:

The following sections 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 the 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 hostname, domain name, or IP address to ensure that everything is resolving properly.

Task 4 Change the References to the Network Configuration

You need to 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, if you use mod_plsql, the PlsqlDatabaseConnectString attribute in the dads.conf file
- For Oracle Portal, portal_dads.conf and sqlnet.ora
- For Oracle Forms Services, sqlnet.ora

This is not an exhaustive list. See Chapter 20 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, as shown in Table 3–1. 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

If you use the IP address in the data source definition, change the system data source to use the new IP address for the database. To do so, you use Oracle WebLogic Server Administration Console:

1. Start the Administration Server.
2. Go to the Administration Console.
3. In the Change Center, click **Lock & Edit.**
4. In the Domain Structure section, expand **Services,** then **JDBC**, and select **Data Sources.**

   The Summary of JDBC Data Sources page is displayed.
5. Select the data source you want to change.

   The Settings page is displayed.
6. Select the Connection Pool tab.
7. To change the host name or IP address, modify the **URL** field. (Usually, this field contains the host name.) For example, to change the host name or domain name, enter the new host name or domain name, as shown in the following example:

   ```
   jdbc:oracle:thin:@newhostname.domainname.com:1522/orcl
   ```
8. Click **Save.**
9. Restart the servers that use this data source. (Click the Target tab to see the servers that use this data source.)

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 for 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 *Oracle Fusion Middleware Installation Planning Guide* 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 *Oracle Fusion Middleware Installation Planning Guide* 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 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 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.
3. 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 *Oracle Fusion Middleware Installation Planning Guide* 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
```

Table 14–1 describes support for IPv6 by Oracle Fusion Middleware components. In the table:

- The column **IPv6 Only** shows whether a component supports using IPv6 only for all communication.
- The column **Dual Stack** shows whether a component supports using both IPv6 and IPv4 for communication. For example, some components do not support using IPv6 only, because some of the communication is with the Oracle Database,

which supports IPv4, not IPv6. Those components support dual stack, allowing for IPv6 communication with other components.

Table 14–1 Support for IPv6

Component	IPv6 Only	Dual Stack	Notes
Oracle WebLogic Server	Yes	Yes	Most Oracle WebLogic Server plug-ins do not support IPV6. IPv6 is enabled with Oracle HTTP Server with the mod_wl_ohs plug-in.
Oracle HTTP Server	Yes	Yes	To configure for IPv6, see Section 14.5.2.
Oracle Web Cache	Yes	Yes	Enabled by default. To disable, see Section 14.5.3.
Oracle SOA Suite	No	Yes	Requires a dual stack, because Oracle Database requires IPv4 addresses.
Oracle WebCenter	No	Yes	Requires a dual stack, because Oracle Database requires IPv4 addresses.
ADF	Yes	Yes	
Oracle Directory Integration Platform	Yes	Yes	Uses JNDI to communicate with LDAP servers and uses data sources to communicate with the database. JNDI and data sources (JDBC) support IPV6. No additional configuration is necessary.
Oracle Directory Services Manager	Yes	Yes	Uses JNDI to communicate with LDAP servers and uses data sources to communicate with the database. JNDI and data sources (JDBC) support IPV6. No additional configuration is necessary.
Oracle Identity Federation	No	Yes	Requires a dual stack, because Oracle Database requires IPv4 addresses.
Oracle Internet Directory	No	Yes	Requires a dual stack, because Oracle Database requires IPv4 addresses. See "Managing IP Addresses" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.
Oracle Platform Security Services	No	Yes	Requires a dual stack, because Oracle Database requires IPv4 addresses.
Oracle Virtual Directory	No	Yes	Requires a dual stack, because Oracle Database requires IPv4 addresses. See *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*.
Oracle Single Sign-On Server	No	No	Uses Oracle HTTP Server proxy, which can be configured for IPv6. Oracle Single Sign-On must be Release 10.1.4.3. See Section 14.5.4.
Oracle Portal	No	No	Uses Oracle HTTP Server reverse proxy to communicate with Oracle Web Cache or Oracle HTTP Server, which can be configured for IPv6. See "Configuring Reverse Proxy Servers" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal* for more information.

Table 14–1 (Cont.) Support for IPv6

Component	IPv6 Only	Dual Stack	Notes
Oracle Forms Services	No	No	Uses reverse proxy to communicate with Oracle Web Cache or Oracle HTTP Server, which can be configured for IPv6.
Oracle Reports	No	No	Uses reverse proxy to communicate with Oracle Web Cache or Oracle HTTP Server, which can be configured for IPv6.
Oracle Business Intelligence Discoverer	No	No	Uses reverse proxy to communicate with Oracle Web Cache or Oracle HTTP Server, which can be configured for IPv6.

The following topics provide more information about Oracle Fusion Middleware support for IPv6:

- Supported Topologies for IPv6 Network Protocols
- Configuring Oracle HTTP Server for IPv6
- Disabling IPv6 Support for Oracle Web Cache
- Configuring Oracle Single Sign-On to Use Oracle HTTP Server with IPv6
- Configuring Oracle Access Manager Support for IPv6

14.5.1 Supported Topologies for IPv6 Network Protocols

The following topologies for IPv4 and IPv6 are supported (dual-stack means that the host is configured with both IPv4 and IPv6):

- Topology A:
  - Oracle Database on IPv4 protocol host
  - Oracle WebLogic Server on dual-stack host
  - Clients on IPv4 protocol host
  - Clients on IPv6 protocol host
- Topology B:
  - Oracle Database on IPv4 protocol host
  - One or more of the following components on dual-stack hosts: Oracle WebLogic Server, Oracle SOA Suite, Oracle WebCenter, Oracle Business Activity Monitoring, Fusion Middleware Control
  - Oracle HTTP Server with mod_wl_ohs on IPv6 protocol host
- Topology C:
  - Database, such as MySQL, that supports IPv6 on IPv6 protocol host
  - Oracle WebLogic Server on IPv6 protocol host
  - Clients on IPv6 protocol host
- Topology D:
  - Oracle Database on IPv4 protocol host

  - One or more of the following components on dual-stack hosts: Identity Management, Oracle SOA Suite, Oracle WebCenter, Oracle Business Activity Monitoring, Fusion Middleware Control
  - Clients on IPv4 protocol host
  - Clients on IPv6 protocol host
- Topology E:
  - Oracle Database on IPv4 protocol host
  - One or more of the following components on IPv4 protocol host: Oracle Portal, Oracle Forms Services, Oracle Reports, Oracle Business Intelligence Discoverer, and Oracle Single Sign-On Release 10.1.4.3
  - Oracle HTTP Server with mod_proxy on dual-stack host
  - Clients on IPv6 protocol host
- Topology F:
  - Oracle Access Manager Release 10.1.4.3 and applications, such as SOA composite applications, on IPv4 protocol host
  - Oracle HTTP Server with mod_proxy on dual-stack host
  - Clients on IPv6 protocol host
- Topology G:
  - Oracle Database on IPv4 protocol host
  - One or more of the following components on IPv4 protocol host: Oracle SOA Suite, Oracle WebCenter, Oracle Business Activity Monitoring, Fusion Middleware Control on IPv4 protocol host
  - Oracle HTTP Server with mod_wl_ohs on dual-stack host
  - Clients on IPv6 protocol host

14.5.2 Configuring Oracle HTTP Server for IPv6

To configure Oracle HTTP Server to communicate using IPv6, you modify configuration files in the following directory:

```
ORACLE_INSTANCE/config/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:

```
ORACLE_INSTANCE/config/OHS/ohs_name
```

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 WLLogFile path. Oracle HTTP Server logs module-specific messages.

Note the following limitations:

- Dynamic clusters are supported only on IPv4 nodes, or in a mixed cluster where each node is configured with a resolvable host name (instead of an IP address or a blank) in the Listen Address.

  To change the Listen Address, use the Oracle WebLogic Server Administration Console and edit the Listen Address in the Server: Configuration: General page, as described in the Oracle WebLogic Server Administration Console help.

- If the cluster contains IPv6 nodes and the host names are not resolvable, the cluster must be static, not dynamic. To set the cluster to static, change the DynamicServerList to Off. If you add or delete any cluster members, you must manually update the configuration file and restart Oracle HTTP Server.

  To change the DynamicServerList to Off, edit the Oracle HTTP Server configuration files.

14.5.3 Disabling IPv6 Support for Oracle Web Cache

By default, IPv6 support is enabled for Oracle Web Cache. You can disable it in the webcache.xml file, which is located in the following directory:

```
(UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
(Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
```

In the file, change the value of the IPV6 element to "NO". For example:

```
<IPV6 ENABLED="NO"/>
```

If the IPV6 element does not exist in the webcache.xml file, you can add the element to the file. Add it after the MULTIPORT element, as shown in the following example:

```
...
         <LISTEN IPADDR="ANY" PORT="7786" PORTTYPE="ADMINISTRATION"/>
         <LISTEN IPADDR="ANY" PORT="7788" PORTTYPE="INVALIDATION"/>
         <LISTEN IPADDR="ANY" PORT="7787" PORTTYPE="STATISTICS"/>
      </MULTIPORT>
      <IPV6 ENABLED="NO"/>
```

14.5.4 Configuring Oracle Single Sign-On to Use Oracle HTTP Server with IPv6

Oracle Single Sign-On Server supports IPv4. However, you can configure Oracle Single Sign-On Server to work with clients that support IPv6 by setting up a proxy server and a reverse proxy.

The steps in this section assume that you have installed Oracle Single Sign-On Server Release 10.1.4.3 and a proxy server such as Oracle HTTP Server that acts as a front end to the Oracle Single Sign-On Server.

Take the following steps to configure Oracle Single Sign-On to work with clients that support IPv6:

1. Enable the proxy server:

   a. Run the ssocfg script on the single sign-on middle tier. This script changes the host name stored in the single sign-on server to the proxy host name. Use the following command syntax, entering values for the protocol, host name, and port of the proxy server:

   ```
   (UNIX) $ORACLE_HOME/sso/bin/ssocfg.sh http proxy_server_name proxy_port
   (Windows) %ORACLE_HOME%\sso\bin\ssocfg.bat http proxy_server_name proxy_
   ```

```
port
```

b. Update the targets.xml file on the single sign-on middle tier. The file is located in:

```
(UNIX) ORACLE_HOME/sysman/emd
(Windows) ORACLE_HOME\sysman\emd
```

Open the file and find the target type `oracle_sso_server`. Within this target type, locate and edit the three attributes that you passed to ssocfg:

- HTTPMachine—the HTTP server host name
- HTTPPort—the SSL port number of the Oracle HTTP server
- HTTPProtocol—the server protocol

c. Add the lines that follow to the httpd.conf file on the single sign-on middle tier. The file is at ORACLE_HOME/Apache/Apache/conf. These lines change the directive ServerName from the name of the actual server to the name of the proxy:

```
KeepAlive off
ServerName proxy_host_name
Port proxy_port
```

Note that if you are using SSL, the port must be an SSL port such as 4443.

d. (SSL only) If you have configured SSL communication between just the browser and the proxy server, configure mod_certheaders on the middle tier. This module enables the Oracle HTTP Server to treat HTTP proxy requests that it receives as SSL requests. Add the lines that follow to httpd.conf. You can place them at the end of the file; where they appear is unimportant.

Enter this line to load the module:

```
(UNIX) LoadModule certheaders_module libexec/mod_certheaders.so
(Windows) LoadModule certheaders_module modules/ApacheModuleCertHeaders.dll
```

If you are using Oracle Web Cache as a proxy, enter this line:

```
AddCertHeader HTTPS
```

If you are using a proxy other than Oracle Web Cache, enter this line:

```
SimulateHttps on
```

e. Reregister mod_osso on the single sign-on middle tier. This step configures mod_osso to use the proxy host name instead of the actual host name. For example, on Linux:

```
$ORACLE_HOME/sso/bin/ssoreg.sh
   -oracle_home_path ORACLE_HOME
   -site_name example.mydomain.com
   -config_mod_osso TRUE
   -mod_osso_url http://example.mydomain.com
```

f. Update the Distributed Configuration Management schema:

```
ORACLE_HOME/dcm/bin/dcmctl updateconfig
```

g. Restart the single sign-on middle tier:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

h. Log in to the single sign-on server, using the single sign-on login URL:

```
http://proxy_host_name:proxy_port/sso/
```

This URL takes you to the single sign-on home page. If you are able to log in, you have configured the proxy correctly.

2. If you have not already done so, install Oracle HTTP Server 11*g* Release 1 (11.1.1) to use as a reverse proxy for IPv6.
3. Change the Oracle HTTP Server 11*g* Release 1 (11.1.1) configuration to enable reverse proxy:

a. Stop Oracle HTTP Server:

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

b. Edit the following file:

```
(UNIX) ORACLE_INSTANCE/config/OHS/ohs_name/httpd.conf
(Windows) ORACLE_INSTANCE\config\OHS\ohs_name\httpd.conf
```

Append the following to the httpd.conf file:

```
#---Added for Mod Proxy
ProxyRequests Off

<Proxy *>
Order deny,allow
Allow from all
</Proxy>

ProxyPass /sso http://OHS_host:OHS_port/sso
ProxyPass / http://OHS_host:OHS_port/
ProxyPassReverse / http://OHS_host:OHS_port/
ProxyPreserveHost On
```

In the example, *OHS_host* and *OHS_port* are the host name and port of the front-end server for Oracle Single Sign-On, discussed in Step 1.

c. Restart the Oracle HTTP Server. For example, to restart ohs1:

```
opmnctl startproc ias-component=ohs1
```

14.5.5 Configuring Oracle Access Manager Support for IPv6

Oracle Access Manager supports Internet Protocol Version 4 (IPv4). Oracle Fusion Middleware supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6). IPv6 is enabled with Oracle HTTP Server with the mod_wl_ohs plug-in.

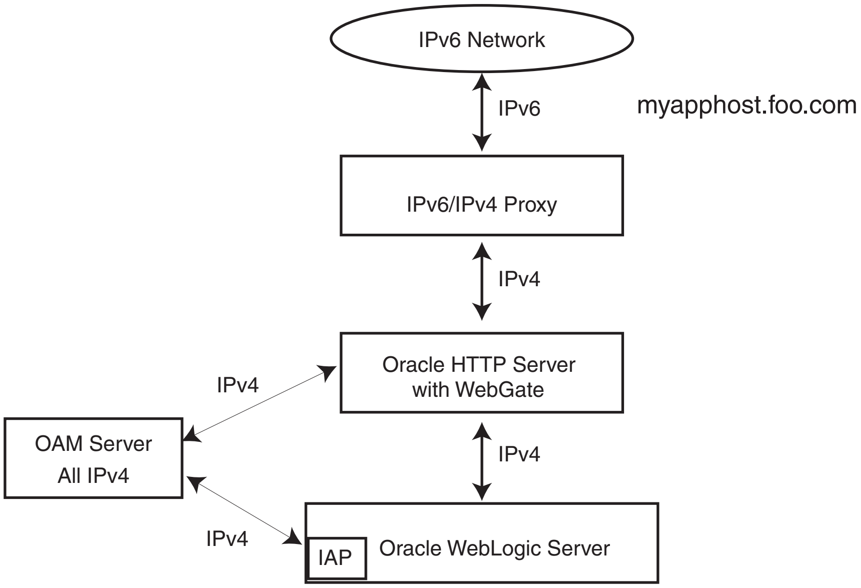
You can configure Oracle Access Manager to work with clients that support IPv6 by setting up a reverse proxy server. Several scenarios are provided here. Be sure to choose the right configuration for your environment.

14.5.5.1 Simple Authentication with IPv6

Figure 14–1 illustrates simple authentication with Oracle Access Manager configured to use the IPv6/IPv4 proxy.

Note: In a WebGate profile, an IPv6 address cannot be specified. In a WebGate profile, the virtual host name must be specified as a host name, for example, *myapphost.foo.com*, not as an IP address.

Figure 14–1 Simple Authentication with the IPv6/IPv4 Proxy



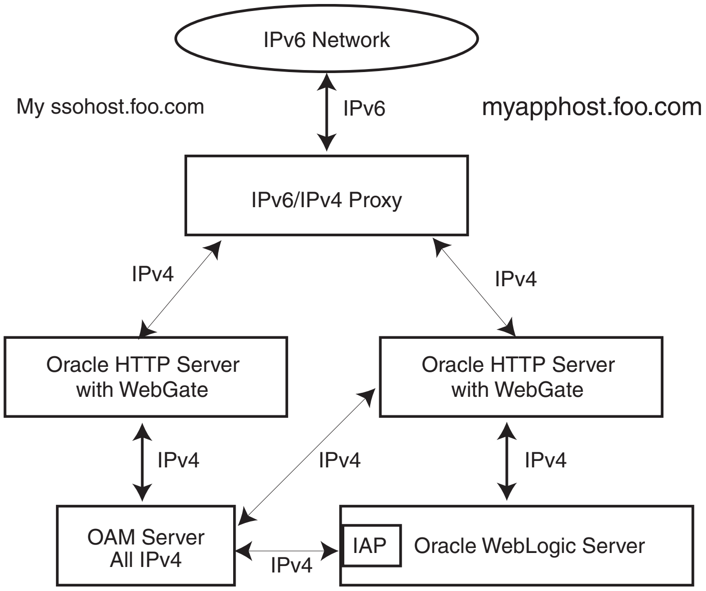
As illustrated in Figure 14–1, the IPv6 network communicates with the IPv6/IPv4 proxy, which in turn communicates with the Oracle HTTP Server and WebGate using IPv4. WebGate, Oracle Access Manager servers, and Oracle WebLogic Server with the Authentication provider all communicate with each other using IPV4.

14.5.5.2 Configuring IPv6 with an Authenticating WebGate and Challenge Redirect

Figure 14–2 illustrates configuration with a single IPv6 to IPv4 proxy (even though *myssohost* and *myapphost* could use separate proxies).

Note: In a WebGate profile, the virtual host name must be specified as a host name, for example, *myapphost.foo.com*, not as an IP address. The redirect host name, for example, *myssohost.foo.com* must also be specified as a host name and not an IP address. The IPv6 address cannot be specified in a WebGate profile.

Figure 14–2 IPv6 with an Authenticating WebGate and Challenge Redirect



As illustrated in Figure 14–2, the IPv6 network communicates with the IPv6/IPv4 proxy, which in turn communicates with the Oracle HTTP Server using IPv4. WebGate, Oracle Access Manager server, and Oracle WebLogic Server with the Identity Asserter all communicate with each other using IPV4.

You should be able to access the application from a browser on the IPv4 network directly to the IPv4 server host name and have login with redirect to IPv6 *myssohost.foo*.com.

14.5.5.3 Considerations

The following considerations apply to each intended usage scenario:

- IP validation does not work by default. To enable IP validation, you must add the IP address of the Proxy server as the WebGate's IPValidationException parameter value in the Access System Console.
- IP address-based authorization does not work because all requests come through one IP (proxy IP) that would not serve its purpose.

14.5.5.4 Prerequisites

Regardless of the manner in which you plan to use Oracle Access Manager with IPv6 clients, the following tasks should be completed before you start:

- Install an Oracle HTTP Server instance to act as a reverse proxy to the Web server (required for WebGate).
- Install and complete the initial set up of Oracle Access Manager (Identity Server, WebPass, Policy Manager, Access Server, WebGate) as described in *Oracle Access Manager Access Administration Guide*.

See Also:

- *Oracle Fusion Middleware Installation Guide for Web Tier*
- *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server*

14.5.5.5 Configuring IPv6 with Simple Authentication

Use the procedure in this section to configure your environment for simple authentication with Oracle Access Manager using the IPv6/IPv4 proxy. See Figure 14–1 for a depiction of this scenario.

The configuration in this procedure is an example only. In the example, *OHS_host* and *OHS_port* are the host name and port of the actual Oracle HTTP Server with WebGate. You must use values for your environment.

> **Note:** For this configuration you must use the Web server on which the WebGate is deployed as the Preferred HTTP host in the WebGate profile. You cannot use the IPv6 proxy name.

To configure IPv6 with simple authentication:

1. Configure Oracle HTTP Server 11g Release 1 (11.1.1) or any other server to enable reverse proxy:

   a. Stop Oracle HTTP Server with the following command:

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

   b. Edit the following file:

   ```
   (UNIX) ORACLE_INSTANCE/config/OHS/ohs_name/httpd.conf
   (Windows) ORACLE_INSTANCE\config\OHS\ohs_name\httpd.conf
   ```

   c. Append the following to the httpd.conf file:

   ```
   #---Added for Mod Proxy
   <IfModule mod_proxy.c>

   ProxyRequests Off
   ProxyPreserveHost On

   ProxyPass /http://OHS_host:OHS_port/
   ProxyPassReverse /http://OHS_host:OHS_port/

   </IfModule>
   ```

   d. Restart Oracle HTTP Server using the following command:

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

2. Log in to the Access System Console. For example:

   ```
   http://hostname:port/access/oblix
   ```

   In the example, *hostname* refers to computer that hosts the WebPass Web server; *port* refers to the HTTP port number of the WebPass Web server instance; /access/oblix connects to the Access System Console.

   The Access System main page appears.

3. Click **Access System Configuration**, and then click **AccessGate Configuration**.

   The Search for AccessGates page appears. The Search list contains a selection of attributes that can be searched. Remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

4. Select the search attribute and condition from the lists (or click **All** to find all AccessGates), and then click **Go.**
5. Click an AccessGate's name to view its details.
6. Click **Modify.**
7. For **Preferred HTTP Host**, specify the Web server name on which WebGate is deployed as it appears in all HTTP requests. The host name within the HTTP request is translated into the value entered into this field regardless of the way it was defined in a user's HTTP request.
8. To enable IP validation, add the IP address of the proxy server as the value of the **IPValidationException** parameter.
9. Click **Save.**

14.5.5.6 Configuring IPv6 with an Authenticating WebGate and Challenge Redirect

Use the procedure in this section to configure your environment to use Oracle Access Manager with the IPv6/IPv4 proxy and an authenticating WebGate and challenge redirect. Figure 14–2 shows a depiction of this scenario.

The following procedure presumes a common proxy for both form-based authentication and the resource WebGate. For example, suppose you have the following configuration:

- Resource WebGate is installed on http://*myapphostv4.foo*.com/
- Resource is on http://*myapphostv4.foo*.com/testing.html
- Authenticating WebGate is on http://*myssohostv4.foo*.com/
- Login form is http://*myssohostv4.foo*.com/oamsso/login.html
- Reverse proxy URL is http://*myapphost.foo*.com/

> **Note:** For this configuration, the Preferred HTTP host must be the name of the Oracle HTTP Server Web server that is configured for this WebGate. For example, a WebGate deployed on *myapphost4.foo*.com must use *myapphost4.foo*.com as the Preferred HTTP host. You cannot use the IPv6 proxy name.

In the following procedure, you configure the Oracle HTTP Server, configure WebGate profiles to use the corresponding Oracle HTTP Server as the Preferred HTTP host, and configure the form-based authentication scheme with a challenge redirect value of the reverse proxy server URL (`http://myapphost.foo.com/` in this example).

Be sure to use values for your own environment.

To configure IPv6 with an authenticating WebGate and challenge redirect:

1. Configure Oracle HTTP Server 11g Release 1 (11.1.1) or any other server, as follows:

   a. Stop Oracle HTTP Server with the following command:

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

   b. Edit the following file:

   ```
   UNIX: ORACLE_INSTANCE/config/OHS/ohs_name/httpd.conf
   ```

```
Windows: ORACLE_INSTANCE\config\OHS\ohs_name\httpd.conf
```

c. Append the following information for your environment to the httpd.conf file. For example:

```
<IfModule mod_proxy.c>
ProxyRequests On
ProxyPreserveHost On
#Redirect login form requests and redirection requests to Authentication
WebGate

ProxyPass /obrareq.cgi   http://myssohostv4.foo.com/obrareq.cgi
ProxyPassReverse /obrareq.cgi  http://myssohostv4.foo.com/obrareq.cgi

ProxyPass /oamsso/login.html http://myssohostv4.foo.com/oamsso/login.html
ProxyPassReverse /oamsso/login.html http://myssohostv4.foo.com/oamsso/login
.html

ProxyPass /access/sso   http://myssohostv4.foo.com/ /access/sso
ProxyPassReverse /access/sso http://myssohostv4.foo.com/access/sso

# Redirect resource requests to Resource WG
ProxyPass /http://myapphostv4.foo.com /
ProxyPassReverse /http://myapphostv4.foo.com /

</IfModule>
```

d. Restart Oracle HTTP Server using the following command:

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

2. In the Access System Console, set the Preferred HTTP host for each WebGate as follows:

   a. Log in to the Access System Console. For example:

   ```
   http://hostname:port/access/oblix
   ```

   In the example, *hostname* refers to computer that hosts the WebPass Web server; *port* refers to the HTTP port number of the WebPass Web server instance; /access/oblix connects to the Access System Console.

   The Access System main page appears.

   b. Click **Access System Configuration**, and then click **AccessGate Configuration**.

   The Search for AccessGates page appears. The Search list contains a selection of attributes that can be searched. Remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

   c. Select the search attribute and condition from the lists (or click **All** to find all AccessGates), and then click **Go.**

   d. Click an AccessGate's name to view its details.

   e. Click **Modify.**

   f. For **Preferred HTTP Host,** specify the name of the Oracle HTTP Server Web server that is configured for this WebGate. For instance, a WebGate deployed on *myapphostv4.foo*.com must use *myapphostv4.foo*.com as the Preferred HTTP host.

g. To enable IP validation, add the IP address of the Proxy server as the value of the **IPValidationException** parameter.

h. Click **Save.**

i. Repeat for each WebGate and specify name of the Oracle HTTP Server Web server that is configured for this WebGate.

3. From the Access System Console, modify the Form authentication scheme to include a challenge redirect to the Proxy server, as follows:

   a. Click **Access System Configuration**, and then click **Authentication Management**.

   b. Click the name of the scheme to modify, and then click **Modify.**

   c. Configure the challenge redirect value to the Proxy server URL. In this example, the Proxy server URL is `http://myapphost.foo.com/`

   d. Click **Save.**

14.5.5.7 Configuring IPv6: Separate Proxy for Authentication and Resource WebGates

Use the procedure in this section to configure a separate proxy for authentication and resource WebGates. In this configuration, you have multiple proxies: for example a separate proxy for the authentication WebGate and another proxy for the resource WebGate. You can access the application from a browser on the IPv4 network directly to an IPv4 server host name with a login redirect to an IPv6 host. For example:

- Resource WebGate is on http://*myapphostv4.foo*.com/
- Authenticating WebGate is on http://*myssohostv4.foo*.com
- Proxy used for *myapphostv4.foo*.com should be *myapphostv4.foo*.com
- Proxy used for *myssohostv4.foo*.com should be *myssohostv4*.com

> **Note:** You cannot use the IPv6 proxy name as the Preferred HTTP host in a WebGate profile.

In the example, *OHS_host* and *OHS_port* are the host name and port of the actual Oracle HTTP Server that is configured for WebGate. Be sure to use values for your own environment.

To configure IPv6 with a separate proxy for authentication and resource WebGates:

1. Configure Oracle HTTP Server 11g Release 1 (11.1.1) or any other server for multiple proxies, as follows:

   a. Stop Oracle HTTP Server with the following command:

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

   b. Edit the following file:

   ```
   UNIX: ORACLE_INSTANCE/config/OHS/ohs_name/httpd.conf
   Windows: ORACLE_INSTANCE\config\OHS\ohs_name\httpd.conf
   ```

   c. Append the following information for your environment to the httpd.conf file. For example:

   ```
   <IfModule mod_proxy.c>
   ```

```
ProxyRequests Off
ProxyPreserveHost On

ProxyPass /http://OHS_host:OHS_port
ProxyPassReverse /http://OHS_host:OHS_port

</IfModule>
```

   d. Restart Oracle HTTP Server using the following command:

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

2. In the Access System Console, set the Preferred HTTP host for each WebGate as follows:

   a. Log in to the Access System Console. For example:

      ```
      http://hostname:port/access/oblix
      ```

      In the example, *hostname* refers to computer that hosts the WebPass Web server; *port* refers to the HTTP port number of the WebPass Web server instance; /access/oblix connects to the Access System Console.

      The Access System main page appears.

   b. Click **Access System Configuration**, and then click **AccessGate Configuration**.

      The Search for AccessGates page appears. The Search list contains a selection of attributes that can be searched. Remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

   c. Select the search attribute and condition from the lists (or click **All** to find all AccessGates), and then click **Go.**

   d. Click an AccessGate's name to view its details.

   e. Click **Modify.**

   f. For **Preferred HTTP Host,** specify the name of the Oracle HTTP Server Web server that is configured for this WebGate. For instance, a WebGate deployed on *myapphostv4.foo*.com must use *myapphostv4.foo*.com as the Preferred HTTP host.

   g. To enable IP validation, add the IP address of the Proxy server as the value of the **IPValidationException** parameter.

   h. Click **Save.**

   i. Repeat for each WebGate and specify the name of the Oracle HTTP Server Web server that is configured for this WebGate.

3. From the Access System Console, modify the Form authentication scheme to include a challenge redirect to the Proxy server, as follows:

   a. Click **Access System Configuration**, and then click **Authentication Management**.

   b. Click the name of the scheme to modify, and then click **Modify.**

   c. Configure the challenge redirect value to the Proxy server URL that acts as a reverse proxy for the authentication WebGate. In this example, the Proxy server URL is `http://myssohost.foo.com/`.

   d. Click **Save.**

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:

15

Introducing Backup and Recovery

This chapter provides an introduction to backing up and recovering Oracle Fusion Middleware.

This chapter includes the following topics:

- 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

An Oracle Fusion Middleware environment can consist of different components and configurations. A typical Oracle Fusion Middleware environment contains an Oracle WebLogic Server domain with Java components, such as Oracle SOA Suite, and an Oracle WebLogic Server domain with Identity Management components. It can also include one or more Oracle instances.

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, you might update configuration files in the installation. 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:

- Section 2.2 for conceptual information about an Oracle WebLogic Server domain
- Section 2.2.1 for conceptual information about the Administration Server
- Section 2.2.2 for conceptual information about Managed Servers and clusters
- Section 2.2.3 for conceptual information about 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. Then, 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

Managed Servers maintain 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

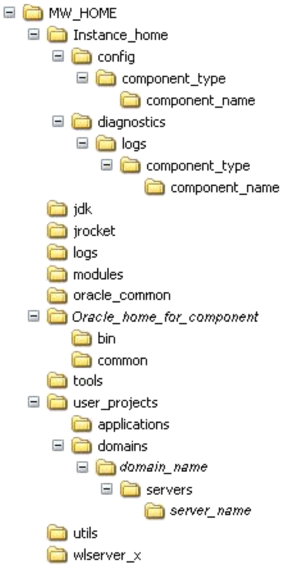
Configuration changes are updated in a Managed Server during the following events:

- On each Managed Server restart, the latest configuration is retrieved from the Administration Server. This happens even when the 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 like configuration changes, deploy or redeploy 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:



```
MW_HOME
  Instance_home
    config
      component_type
        component_name
    diagnostics
      logs
        component_type
          component_name
  jdk
  jrocket
  logs
  modules
  oracle_common
  Oracle_home_for_component
    bin
    common
  tools
  user_projects
    applications
    domains
      domain_name
        servers
          server_name
  utils
  wlserver_x
```

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, or jar.

  For example, for online backups on Windows, use copy; for offline backups on Windows, use copy, xcopy, or jar. Do not use Winzip because it does not work with long filenames or extensions.

  For example, for Linux and UNIX, use tar.

  Ensure that the tool you are using preserves the permissions of the files.

- Oracle Recovery Manager (RMAN) to back up database-based metadata repositories.

- Oracle WebLogic Server Pack and Unpack Utility.

  The pack command creates a template archive (.jar) file that contains a snapshot of either an entire domain or a subset of a domain. You can use a template that contains a subset of a domain to create a Managed Server domain directory hierarchy on a remote computer.

  Alternatively, you can use a template that contains an entire domain to create the domain on a remote computer.

> **See Also:** *Oracle Fusion Middleware Creating Templates and Domains Using the Pack and Unpack Commands*

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 backup 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, and all system components in the Oracle instances 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 3.4.2.

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.

Static files and directories are those that do not change frequently. These include:

- The Middleware home (MW_HOME). MW_HOME consists of a WebLogic Server home, and optionally a Oracle Common home and an Oracle home. It can also contain the user_projects directories, which contains Oracle WebLogic Server domains, and Oracle instance homes, which are not static files.
- OraInventory
- OraInst.loc and oratab files, which are 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 Web Cache, 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 MW_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 domain directories separately because the Administration Server contains information about all of the Managed Servers in its domain.
- All Oracle instance homes, which reside, by default, in the MW_HOME but can be configured to be in a different location
- Application artifacts, such as .ear or .war files

  You do not need to backup application artifacts in a Managed Server domain 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. However, note the limitation described in Section 16.2.

15.3.2 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 15.3.1. Perform an offline full backup at the following times:
  - Immediately after you install Oracle Fusion Middleware
  - Immediately after an operating system software upgrade
- **Perform an online backup of run-time artifacts:** This involves backing up the run-time artifacts described in Section 15.3.1. 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 or cluster.
  - After making configuration changes to a component or cluster.
  - 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 an offline backup of static files and directories:** This involves backing up the static files and directories described in Section 15.3.1. Perform an offline backup of static files and directories at the following times:
  - After patching your Oracle Fusion Middleware environment. This backup serves as the basis for subsequent online backups.
  - After upgrading your Oracle Fusion Middleware environment. This backup serves as the basis for subsequent online backups.

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
- Metadata repository files
- Oracle system files
- Windows Registry key
- 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.

  For example, for online recovery on Windows, use copy; for offline recovery on Windows, use copy, xcopy, or jar. Do not use Winzip because it does not work with long filenames or extensions.

  For example, for Linux and UNIX, use tar.

- 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:

- A domain
- The WebLogic Server Administration Server
- A Managed Server
- The Middleware home
- An Oracle instance home
- A component, such as Oracle HTTP Server or Oracle Web Cache

- A 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 an Oracle instance home or a component, 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 SOA Suite
- Backup and Recovery Recommendations for Oracle WebCenter
- Backup and Recovery Recommendations for Oracle Identity Management
- Backup and Recovery Recommendations for Oracle JRF Installations
- Backup and Recovery Recommendations for Web Tier Installations
- Backup and Recovery Recommendations for Oracle Portal, Oracle Forms Services, and Oracle Reports Installations

These topics include information about configuration files for particular components. Note that the list of files in not an exhaustive list. You do not back up or recover the individual files. Generally, you back up or recover a Middleware home, the domain, Managed Server, Oracle home, or Oracle instance.

For the steps you take to back up your environment, see Section 16.3. For the steps you take to recover a component, see Chapter 17.

15.5.1 Backup and Recovery Recommendations for Oracle SOA Suite

The following sections describe backup and recovery recommendations for Oracle SOA Suite:

- Backup and Recovery Recommendations for Oracle BPEL Process Manager
- Backup and Recovery Recommendations for Oracle Business Activity Monitoring
- Backup and Recovery Recommendations for Oracle B2B
- Backup and Recovery Recommendations for Oracle Business Rules
- Backup and Recovery Recommendations for Oracle WebLogic Server JMS

For the steps you need to take to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.5.

15.5.1.1 Backup and Recovery Recommendations for Oracle BPEL Process Manager

This section describes the Oracle BPEL Process Manager data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.5.

Configuration Files

Configuration files are stored in the database.

Database Repository Dependencies

Process definition and configuration files are stored in the MDS schema. The dehydration store is stored in the BPEL schema.

Backup Recommendations

Back up the Administration Server domain directories. Back up the database after any configuration changes, including changes to global fault policies, callback classes for workflows and resource bundles that can potentially be outside the suitcase. Also back up the database after deploying a new composite or redeploying a composite.

Recovery Recommendations

Recover the database to the most recent point in time, if needed. Point-in-time recovery ensures that the latest process definitions and in-flight instances are restored. However, this may result in reexecution of the process steps. Oracle recommends that you strive for idempotent Oracle BPEL Process Manager processes. If the system contains processes that are not idempotent, you must clean them up from the dehydration store before starting Oracle Fusion Middleware. See *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite* for more information.

Because instances obtain the process definition and artifacts entirely from the database, there is no configuration recovery needed after the database is recovered to the most current state; instances should continue to function correctly.

For redeployed composites, a database recovery ensures consistency between the dehydrated in-flight processes and their corresponding definition since the process definition is stored in database repository where dehydrated instances are also stored.

15.5.1.2 Backup and Recovery Recommendations for Oracle Business Activity Monitoring

This section describes the Oracle Business Activity Monitoring data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.5.

Configuration Files

```
MW_HOME/SOA_ORACLE_HOME/bam
DOMAIN_HOME/config/fmwconfig/servers/AdminServer/adml/server-oracle_
bamweb-11.0.xml
DOMAIN_HOME/config/fmwconfig/servers/AdminServer/adml/server-oracle_
bamserver-11.0.xml
```

```
DOMAIN_HOME/config/fmwconfig/servers/bam-server-name/adml/server-oracle_
bamweb-11.0.xml
DOMAIN_HOME/config/fmwconfig/servers/bam-server-name/adml/server-oracle_
bamserver-11.0.xml
```

Database Repository Dependencies

ORABAM schema.

Backup Recommendations

Back up the Middleware home, the Administration Server domain, the Managed Server domain, and the database containing the ORABAM schema.

Recovery Recommendations

Recover the Managed Server or the Middleware home, or both, depending on the extent of failure.

Recover the database to the most recent point in time, if needed.

15.5.1.3 Backup and Recovery Recommendations for Oracle B2B

This section describes the Oracle B2B data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.5.

Configuration Files

```
DOMAIN_HOME/config/soa-infra/configuration/b2b-config.xml
```

Database Repository Dependencies

MDS schema.

Backup Recommendations

Back up the Administration Server domain, the Oracle home if changes are made to the Oracle B2B configuration file, and the database containing the MDS schema.

Recovery Recommendations

Recover the Managed Server where the soa-infra application is deployed.

Recover the database to the most recent point in time, if needed.

After recovery, if the file Xengine.tar.gz is not unzipped, unzip the files. For example:

```
cd B2B_ORACLE_HOME/soa/thirdparty/edifecs
tar xzvf XEngine.tar.gz
```

15.5.1.4 Backup and Recovery Recommendations for Oracle Business Rules

This section describes the Oracle Business Rules data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.5.

Configuration Files

```
DOMAIN_HOME/config/soa-infra/configuration/businessrules-config.xml
```

Database Repository Dependencies

MDS schema.

Backup Recommendations

Back up the Administration Server domain and the database containing the MDS schema.

Recovery Recommendations

Recover the Managed Server where the soa-infra application is deployed.

Recover the database to the most recent point in time, if needed.

15.5.1.5 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.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3.

Configuration Files

```
DOMAIN_HOME/config/jms
```

If a JMS uses file-system accessible stores, the default file-system store is either in a user-configured location that is specified in config.xml, or in the following location:

```
DOMAIN_HOME/servers/server_name/data/store/default
```

Database Repository Dependencies

If a JMS uses a JDBC accessible store, back up the database.

Backup Recommendations

Back up the domain and the JMS file persistent store if it is not located within the domain.

Back up the schema in the database if JDBC-based persistent store is configured. 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 (in the case of database-based persistence) or using a highly available RAID backed storage device (for example, SAN/NAS).
- If, for whatever reason, you need to restore JMS data to previous point in time, there are potential implications. Restoring the system state to a prior 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. If the persistent store is a custom store that is dedicated to JMS use, then you can delete the entire store.

  Use the following procedure **before recovery** to drain messages in the JMS queue after persistent-store recovery to avoid processing duplicate messages:

  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 **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.**
   e. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes.**

  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 destination.
     c. Select **Monitoring,** then **Show Messages**

        Click **Delete All.**
  3. 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.** Select **Insertion Paused At Startup, Production Paused At Startup,** and **Consumption Paused At Startup.**
     d. Click **Save.**
     e. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes.**

  If the store is not dedicated to JMS use, use the Oracle WebLogic Server JMS message management administrative tooling. This tool can perform import, export, move, and delete 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.

- All JMS data should be backed up continuously and synchronously. Do not use asynchronous snapshots:
  – For Oracle Database-based persistence, use RMAN for backups.
  – For file-based persistence, if you are using a SAN/NAS device, then use snapshot mechanism to take a consistent JMS data backup.
  – If you are using a storage device that does not support block-level snapshot capabilities, then you must shutdown the JMS server to be able to take a consistent backup. This is to ensure that the persistence store is not being written to while the copy operation is being performed. In a clustered environment, you can do so by shutting down one server at time, backing it up and restarting it. You also can create a script to perform these operations using WLST.

Recovery Recommendations

Recover the domain.

If the JMS file persistent store is file-based, recover it from backup. If the JMS file persistent store is database-based, recover the database to the most recent point in time, if needed.

15.5.2 Backup and Recovery Recommendations for Oracle WebCenter

The following sections describe backup and recovery recommendations for Oracle WebCenter:

- Backup and Recovery Recommendations for Oracle WebCenter
- Backup and Recovery Recommendations for Oracle WebCenter Portlets
- Backup and Recovery Recommendations for Oracle WebCenter Discussions Server
- Backup and Recovery Recommendations for Oracle WebCenter Wiki and Blog Server
- Backup and Recovery Recommendations for Oracle Content Server

15.5.2.1 Backup and Recovery Recommendations for Oracle WebCenter

This section describes the Oracle WebCenter data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.7.

Configuration Files

All configuration files are bundled in the EAR file, which is located in the domain.

Database Repository Dependencies

WEBCENTER and MDS schemas

Backup Recommendations

Back up the Administration Server domain and the database containing the WEBCENTER and MDS schemas.

Recovery Recommendations

Recover the Oracle WebCenter domain.

Recover the database containing the WEBCENTER and MDS schemas to the most recent point in time, if needed.

15.5.2.2 Backup and Recovery Recommendations for Oracle WebCenter Portlets

This section describes the Oracle WebCenter Portlets data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.7.

Configuration Files

All configuration files are bundled in the EAR file, which is located in the domain.

Database Repository Dependencies

PORTLET

Backup Recommendations

Back up the Administration Server domain and the database containing the PORTLET schema.

Recovery Recommendations

Recover the Oracle WebCenter domain.

Recover the database to the most recent point in time, if needed.

15.5.2.3 Backup and Recovery Recommendations for Oracle WebCenter Discussions Server

This section describes the Oracle WebCenter Discussions server data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.7.

Configuration Files

Some configuration files are either bundled in the EAR file, which is located in the domain or the files are located elsewhere in the domain. Other configuration files are located in:

```
DOMAIN_HOME/fmwconfig/server/server_name/owc_discussions
```

Database Repository Dependencies

JIVE schema.

Backup Recommendations

Back up the Administration Server domain and the database containing the JIVE schema.

Recovery Recommendations

Recover the Oracle WebCenter domain.

Recover the database to the most recent point in time, if needed.

15.5.2.4 Backup and Recovery Recommendations for Oracle WebCenter Wiki and Blog Server

This section describes the Oracle WebCenter Wiki and Blog Server data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.7.

Configuration Files

Configuration files are bundled in the WAR file, which is located in the domain.

Database Repository Dependencies

WIKI schema.

Backup Recommendations

Back up the Administration Server domain and the database containing the WIKI schema.

Recovery Recommendations

Recover the Oracle WebCenter domain.

Recover the database to the most recent point in time, if needed.

15.5.2.5 Backup and Recovery Recommendations for Oracle Content Server

For information about backing up and recovering Oracle Content Server, see *Getting Started with Content Server* which is available at:

http://download.oracle.com/docs/cd/E10316_01/owc.htm

Database Repository Dependencies

OCSERVER schema.

15.5.3 Backup and Recovery Recommendations for Oracle Identity Management

The following sections describe backup and recovery recommendations for Oracle Identity Management:

- Backup and Recovery Recommendations for Oracle Internet Directory
- Backup and Recovery Recommendations for Oracle Virtual Directory
- Backup and Recovery Recommendations for Oracle Directory Integration Platform
- Backup and Recovery Recommendations for Oracle Directory Services Manager
- Backup and Recovery Recommendations for Oracle Identity Federation

15.5.3.1 Backup and Recovery Recommendations for Oracle Internet Directory

This section describes the Oracle Internet Directory data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.9.1.

Configuration Files

```
ORACLE_INSTANCE/config/tnsnames.ora
ORACLE_INSTANCE/OID/admin
ORACLE_INSTANCE/OID/ldap/server/plugin
ORACLE_INSTANCE/OID/component_name
ORACLE_INSTANCE/config/OID/component_name
```

Database Repository Dependencies

ODS and ODSSM schemas.

Backup Recommendations

Back up the Oracle Internet Directory component directory and the Oracle instance home that contains Oracle Internet Directory. Back up the database containing the ODS and ODSSM schemas.

Recovery Recommendations

Recover the Oracle instance home that contains Oracle Internet Directory.

Recover the database to the most recent point in time, if needed.

15.5.3.2 Backup and Recovery Recommendations for Oracle Virtual Directory

This section describes the Oracle Virtual Directory data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.9.2.

Configuration Files

```
ORACLE_INSTANCE/OVD/component_name
ORACLE_INSTANCE/config/OVD/component_name
ORACLE_INSTANCE/diagnostics/logs/OVD/component_name
```

Database Repository Dependencies

None.

Backup Recommendations

Back up the Oracle instance home that contains Oracle Virtual Directory.

Recovery Recommendations

Restore the Oracle instance home that contains Oracle Virtual Directory.

15.5.3.3 Backup and Recovery Recommendations for Oracle Directory Integration Platform

This section describes the Oracle Directory Integration Platform data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.9.3.

Configuration Files

```
MW_HOME/user_projects/domains/domain_name/config/fmwconfig/servers/server_
name/applications/dip_version_number/configuration/dip-config.xml
```

dip-config.xml is part of the Oracle Directory Integration Platform application. It is backed up when you back up the Administration Server domain.

Database Repository Dependencies

ODSSM schema, used by Oracle Internet Directory.

Backup Recommendations

Back up the Administration Server domain directories,the Managed Server domain directories, and Oracle Internet Directory and its dependencies.

Recovery Recommendations

Recover the Managed Server where the Oracle Directory Integration Platform application is deployed.

Recover Oracle Internet Directory.

15.5.3.4 Backup and Recovery Recommendations for Oracle Directory Services Manager

This section describes the Oracle Directory Services Manager data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.9.4.

Configuration Files

Oracle Directory Services Manager, which is the graphical user interface for Oracle Internet Directory and Oracle Virtual Directory, does not have configuration files, but keeps track of host and port information of Oracle Internet Directory and Oracle Virtual Directory in serverlist.txt, which is part of the application .ear file:

```
MW_HOME/user_projects/domains/domain_name/servers/server_name/tmp/_WL_user/odsm_
11.1.1.0.0/nx1i7i/war/WEB-INF/serverlist.txt
```

Database Repository Dependencies

None.

Backup Recommendations

Back up the domain.

Recovery Recommendations

To restore Oracle Directory Services Manager, enter the user name and password to connect to Oracle Internet Directory or Oracle Virtual Directory.

15.5.3.5 Backup and Recovery Recommendations for Oracle Identity Federation

This section describes the Oracle Identity Federation data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.9.5.

Configuration Files

```
DOMAIN_HOME/servers/server_name/stage/OIF/11.1.1.0.0/OIF/configuration
```

Database Repository Dependencies

OIF schema.

Backup Recommendations

Back up the Administration Server domain, the Managed Server domain and the database containing the OIF schema.

Recovery Recommendations

Recover the Managed Server where the Oracle Identity Federation application is deployed.

Recover the database to the most recent point in time, if needed.

15.5.4 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

15.5.4.1 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.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3.

Configuration Files

DOMAIN_HOME/config/fmwconfig/policy-accessor-config.xml

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.

15.5.4.2 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.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.9.1.

Configuration Files

DOMAIN_HOME/config/fmwconfig/jps-config.xml

Database Repository Dependencies

None.

Backup Recommendations

Back up the Administration Server domain.

Recovery Recommendations

Restore the jps-config.xml file.

15.5.5 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
- Backup and Recovery Recommendations for Oracle Web Cache

15.5.5.1 Backup and Recovery Recommendations for Oracle HTTP Server

This section describes the Oracle HTTP Server data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.8.1.

Configuration Files

```
ORACLE_INSTANCE/config/OHS/component_name
ORACLE_INSTANCE/diagnostics/logs/OHS/component_name
```

Database Repository Dependencies

None.

Backup Recommendations

Back up the Oracle instance that contains Oracle HTTP Server.

Recovery Recommendations

Restore the Oracle instance that contains Oracle HTTP Server.

15.5.5.2 Backup and Recovery Recommendations for Oracle Web Cache

This section describes the Oracle Web Cache data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.8.2.

Configuration Files

```
ORACLE_INSTANCE/config/WebCache/component_name
ORACLE_INSTANCE/diagnostics/logs/WebCache/component_name
```

Database Repository Dependencies

None.

Backup Recommendations

Back up the Oracle instance that contains Oracle Web Cache.

Recovery Recommendations

Restore the Oracle instance that contains Oracle Web Cache.

15.5.6 Backup and Recovery Recommendations for Oracle Portal, Oracle Forms Services, and Oracle Reports Installations

The following sections describe backup and recovery recommendations for these components:

- Backup and Recovery Recommendations for Oracle Portal
- Backup and Recovery Recommendations for Oracle Forms Services
- Backup and Recovery Recommendations for Oracle Reports
- Backup and Recovery Recommendations for Oracle Business Intelligence Discoverer

15.5.6.1 Backup and Recovery Recommendations for Oracle Portal

This section describes the Oracle Portal data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.10.1.

Configuration Files

```
MW_HOME/user_projects/domains/domain_name/servers/server_
name/stage/portal/portal/configuration/appConfig.xml
MW_HOME/user_projects/domains/domain_name/servers/server_
name/stage/portal/portal/configuration/portal_dads.conf
MW_HOME/user_projects/domains/domain_name/servers/server_
name/stage/portal/portal/configuration/portal_plsql.conf
MW_HOME/user_projects/domains/domain_name/servers/server_
name/stage/portal/portal/configuration/portal_cache.conf
```

Database Repository Dependencies

PORTAL, PORTAL_DEMO, PORTAL_APP, PORTAL_PUBLIC, AND PORTAL_APPROVAL schemas.

Backup Recommendations

Back up the Administration Server domain, the Managed Server domain, and the Oracle instance containing Oracle Portal, as well as the database containing the schemas.

Recovery Recommendations

Recover the WebLogic Server domain and the Oracle instance containing Oracle Portal, as well as the database containing the schemas.

Recover the database to the most recent point in time, if needed.

15.5.6.2 Backup and Recovery Recommendations for Oracle Forms Services

This section describes the Oracle Forms Services data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.10.4.

Configuration Files

Forms Component:

```
ORACLE_INSTANCE/config/Forms/forms
ORACLE_INSTANCE/Forms/forms
```

Forms Common Component:

```
ORACLE_INSTANCE/config/Forms/frcommon
ORACLE_INSTANCE/Forms/frcommon
```

Forms Java EE application and external files:

```
wls_managed_server/stage/formsapp
```

Database Repository Dependencies

Any user-configured database for Oracle Forms Services applications.

Backup Recommendations

Back up the Administration Server domain, the Managed Server domain, and Oracle instance home where Oracle Forms Services is located.

Recovery Recommendations

Restore the Oracle instance home where Oracle Forms Services is located.

15.5.6.3 Backup and Recovery Recommendations for Oracle Reports

This section describes the Oracle Reports data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.10.3.

Configuration Files

For Reports Server:

```
ORACLE_INSTANCE/config/ReportsServer/server_name/rwserver.conf
ORACLE_INSTANCE/config/ReportsServer/server_name/jdbcpds.conf
ORACLE_INSTANCE/config/ReportsServer/server_name/xmlpds.conf
ORACLE_INSTANCE/config/ReportsServer/server_name/textpds.conf
ORACLE_INSTANCE/config/ReportsServer/server_name/rwnetwork.conf
ORACLE_INSTANCE/config/ReportsServer/server_name/pcscomponent.conf
ORACLE_INSTANCE/config/ReportsServer/server_name/component-logs.xml
ORACLE_INSTANCE/config/ReportsServer/server_name/logging.xml
```

For Oracle Reports Servlet:

```
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/cgicmd.dat
DOMAIN_HOME/servers/server_
namestage/reports/reports/configuration/rwservlet.properties
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/rwserver.conf
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/jdbcpds.conf
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/xmlpds.conf
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/textpds.conf
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/rwnetwork.conf
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/logging.xml
DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/logmetadata.xml
```

For Oracle Reports Bridge:

```
ORACLE_INSTANCE/config/ReportsBridge/bridge_name/rwbridge.conf
ORACLE_INSTANCE/config/ReportsBridge/bridge_name/rwnetwork.conf
ORACLE_INSTANCE/config/ReportsBridge/bridge_name/component-logs.xml
ORACLE_INSTANCE/config/ReportsBridge/bridge_name/loggin.xml
ORACLE_INSTANCE/config/ReportsBridge/bridge_name/pcscomponent.xml
```

For Oracle Reports Tool:

```
ORACLE_INSTANCE/config/ReportsTools/rwbuilder.conf
ORACLE_INSTANCE/config/ReportsTools/rwnetwork.conf
ORACLE_INSTANCE/config/ReportsTools/jdbcpds.conf
ORACLE_INSTANCE/config/ReportsTools/xmlpds.conf
```

```
ORACLE_INSTANCE/config/ReportsTools/textpds.conf
ORACLE_INSTANCE/config/ReportsTools/pcscomponent.xml
ORACLE_INSTANCE/config/ReportsTools/rwservlet.properties
ORACLE_INSTANCE/config/ReportsTools/cgicmd.dat
ORACLE_INSTANCE/config/ReportsTools/component-logs.xml
ORACLE_INSTANCE/config/ReportsTools/logging.xml
```

Other directories and files:

```
ORACLE_INSTANCE/reports/server/*.dat
ORACLE_INSTANCE/reports/cache/
ORACLE_INSTANCE/reports/fonts/
ORACLE_INSTANCE/reports/plugins/resource
ORACLE_INSTANCE/diagnostics/logs/reports/ReportsServer
ORACLE_INSTANCE/diagnostics/logs/reports/ReportsBridge
ORACLE_INSTANCE/diagnostics/logs/reports/ReportsTools
(UNIX) ORACLE_INSTANCE/config/reports/bin/rw*.sh
(Windows) ORACLE_INSTANCE\config\reports\bin\rw*.bat
(UNIX) ORACLE_INSTANCE/config/reports/bin/reports.sh
(Windows) ORACLE_INSTANCE\config\reports\bin\reports.bat
(UNIX) ORACLE_INSTANCE/config/reports/bin/namingservice.sh
(Windows) ORACLE_INSTANCE\config\reports\bin\namingservice.bat
```

Database Repository Dependencies

You can configure Oracle Reports to store job-related information, such as scheduled job data, past job data, or job status data in a database.

Backup Recommendations

Back up the Administration Server domain, the Managed Server domain, and the Oracle instance home where Oracle Reports is located.

If a database is configured for Oracle Reports, back up the database.

Recovery Recommendations

Restore the Oracle instance home where Oracle Reports is located.

If a database is configured for Oracle Reports, recover the database to most recent point in time, if needed.

15.5.6.4 Backup and Recovery Recommendations for Oracle Business Intelligence Discoverer

This section describes the Oracle Business Intelligence Discoverer data that must be backed up and restored.

For the steps to recover components, see Section 17.2.6 and Section 17.3.3. For the steps specific to recovering from loss of host, see Section 17.3.3.10.2.

Configuration Files

```
ORACLE_INSTANCE/config/PreferenceServer/disco-comp-name/pref.txt
ORACLE_INSTANCE/config/PreferenceServer/disco-comp-name/.reg_key.dc
DOMAIN_HOME/servers/server_
name/stage/discoverer/discoverer/configuration/configuration.xml
DOMAIN_HOME/config/config.xml
DOMAIN_HOME/config/fmwconfig/servers/server_name/logging.xml
ORACLE_INSTANCE/diagnostics/logs/PreferenceServer/Discoverer_instance_
name/console*
ORACLE_INSTANCE/diagnostics/logs/PreferenceServer/Discoverer_instance_name/log*
DOMAIN_HOME/servers/server_name/logs/discoverer/diagnostic-*.xml
```

```
DOMAIN_HOME/servers/server_name/logs/discoverer/diagnostics*.xml
DOMAIN_HOME/servers/server_name/logs/discoverer/WLS_DISCO-diagnostic-*.xml
```

Database Repository Dependencies

DISCOVERER and DISCOVERER_PS schemas

Backup Recommendations

Back up the Administration Server domain, the Managed Server domain, and the Oracle BI Discoverer Oracle instance home and the database containing the DISCOVERER and DISCOVERER_PS schemas.

Recovery Recommendations

Restore the Oracle instance that contains Oracle BI Discoverer.

Recover the database to the most recent point in time, if needed.

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.

- File systems and files can only be restored as of the last *good* backup. There is no support for roll forward recovery to the current point in time.
- All of the files required for recovery are maintained within the Middleware home, Oracle instance home, and Oracle Inventory (for loss of host use cases) directories. Generally, as long as no administration changes, such as configuration changes, deployments, redeployments, or patching, have been done since the last backup, it is always safe to restore the file system pertaining to a particular component to a previous point in time using the last *good* backup. Thus, new backups must always performed after any administration changes.
- 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 are running in 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.
- If you have multiple Managed Servers running on different hosts (not in a cluster), the domain should be configured to use an external LDAP for policy store instead of using file-base policy store.
- If the Administration Server is on a different host than the Managed Servers, the domain should be configured to use an external LDAP for the policy store instead of using a file-base policy store.

> **See Also:** If you are using Cold Failover Cluster or Disaster Recovery, refer to the *Oracle Fusion Middleware High Availability Guide* for additional information.

16

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 topics:

- 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 15.3.2, 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 3.4.2.
- 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:

  ```
  MW_HOME/wlserver_n/common/bin/nodemanager.properties
  MW_HOME/wlserver_n/common/bin/wlsifconfig.sh
  MW_HOME/wlserver_n/common/bin/setPatchEnv.sh
  MW_HOME/wlserver_n/common/bin/commEnvg.sh
  ```

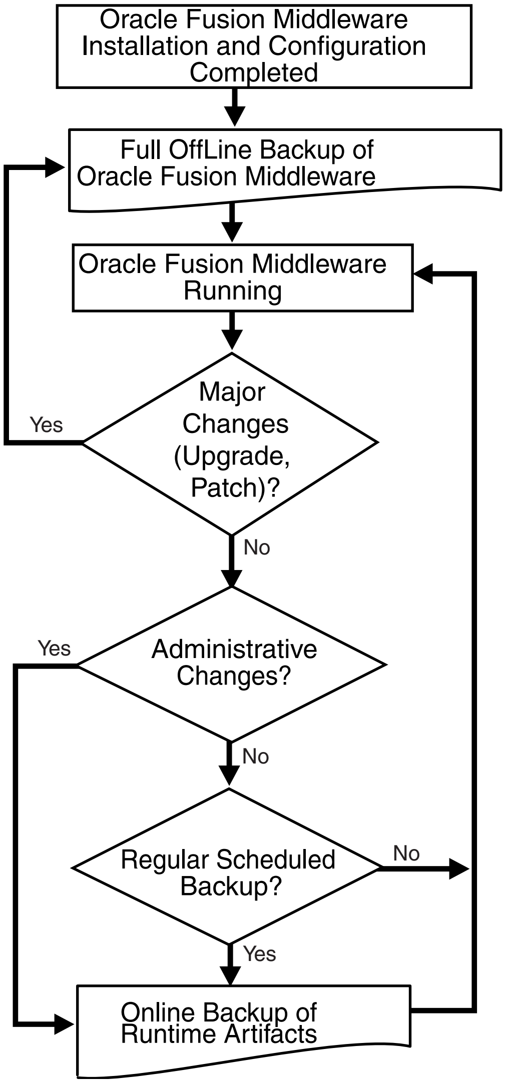
  See Section 16.3.1 for information on performing a full backup.
- Continue to perform backups of run-time artifacts on a regular basis after you establish a new full backup. Oracle recommends that you back up run-time artifacts nightly. See Section 16.3.2 for information on performing a backup of run-time artifacts.
- 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 Middleware home on September 29, 2009, name the backup:

  ```
  mw_home_backup_092909.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


Oracle Fusion Middleware Installation and Configuration Completed
Full OffLine Backup of Oracle Fusion Middleware
Oracle Fusion Middleware Running
Major Changes (Upgrade, Patch)?
Yes
No
Administrative Changes?
Yes
No
Regular Scheduled Backup?
No
Yes
Online Backup of Runtime Artifacts

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.

- Persistent stores: A persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence. For example, it can store persistent JMS (Java Messaging Service) messages or durable subscriber information, as well as temporarily store messages sent to an unavailable destination using the Store-and-Forward feature. The persistent store supports persistence to a file-based store (File Store) or to a JDBC-enabled database (JDBC Store). The default store maintains its data in the data\store\default directory inside the *servername* subdirectory of a domain's root directory.

  It is currently not possible to take consistent backup of persistent stores for a system that uses JMS and transaction logs. This is because the transaction logs can only be file-based and the JMS can be either file-based or it can reside in the database. For highest reliability, use a highly available, fault-tolerant storage (for example, SAN) for JMS and transaction log file stores.

  For clustered servers, Oracle WebLogic Server enables you to migrate a failing server, including the Transaction Recovery Service, to a new system. When the server migrates to another system, it must be able to locate the transaction log records to complete or recover transactions. Transaction log records are stored in the default persistent store for the server.

  If you plan to migrate clustered servers in the event of a failure, you must set up the default persistent store so that it stores records in a shared storage system that is accessible to any potential system to which a failed migratable server might be migrated. For highest reliability, use a shared storage solution (for example, SAN) or an Oracle database that is highly available and supports a point-in-time recovery. This solution is also recommended for all the JMS modules.

- 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 will be uploaded again when you restore the files.

  The default locations for bus stop local files are:

  – For Java components:

    ```
    MW_HOME/user_projects/domains/domain_name/servers/server_
    name/logs/auditlogs/component_type
    ```

  – For system components, such as Oracle HTTP Server or Oracle Internet Directory:

    ```
    ORACLE_INSTANCE/auditlogs/component_type/component_name
    ```

  For more information about Oracle Fusion Middleware Audit Framework and the bus stop, see "Configuring and Managing Auditing" in the *Oracle Fusion Middleware Security Guide*.

16.3 Performing a Backup

You can perform the following types of backups:

- Full offline backup. See Section 16.3.1.
- Online or offline backup of run-time artifacts. See Section 16.3.2.

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 Middleware home, using your preferred tool for archiving. Ensure that the tool you are using preserves the permissions of the files.

For example, for online backups on Windows, use copy; for offline backups on Windows, use copy, xcopy, or jar. Do not use Winzip because it does not work with long filenames or extensions.

For example, for Linux and UNIX, use tar.

The following example shows how to archive and compress the source on UNIX:

```
cd Source_Middleware_Home
tar cf - * | gzip > Middleware_Home.tar.gz
```

The tar utility may issue warnings if the sticky bit is set on some files. You can safely ignore these warnings.

Do not use the jar utility to archive and compress the file system. This avoids warnings or errors from the zip tool about zipping open files (for example, the *ORACLE_HOME*/jdk files).

Take the following steps:

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 3.4.2.
2. Back up the Middleware home (MW_HOME) on all hosts. For example:

   ```
   tar -cf mw_home_backup_092909.tar MW_HOME/*
   ```

3. If the domain is not located within the Middleware home, back up the Administration Server domain separately. This backs up Java components such as Oracle SOA Suite and Oracle WebCenter.

   For example:

   ```
   tar -cf domain_home_backup_092909.tar MW_HOME/user_projects/domains/domain_
   name/*
   ```

   In most cases, you do not need to back up the Managed Server domain directories separately, because the Administration Server domain contains information about the Managed Servers in its domain. See Section 15.5 for information about what you need to back up.

4. If the Oracle instance home is not located within the Middleware home, back up the Oracle instance home. The Oracle instance home contains configuration information about system components, such as Oracle HTTP Server or Oracle Internet Directory. (See Section 3.5.2 for a list of system components.)

   For example:

   ```
   tar -cf sc_home_backup_092909.tar ORACLE_INSTANCE/*
   ```

5. If a Managed Server is not located within the domain, back up the Managed Server directory. For example:

   ```
   tar -cf mg1_home_backup_092909.tar MW_HOME/user_projects/domains/domain_
   name/servers/server_name/*
   ```

6. Back up the OraInventory directory. For example:

   ```
   tar -cf Inven_home_backup_092909 /scratch/oracle/OraInventory
   ```

7. Back up OraInst.loc and oratab files, which are located in the following directory:

   ```
   /etc
   ```

8. Back up the database repositories using the Oracle Recovery Manager (RMAN). For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*, which is available at:

   http://www.oracle.com/technology/documentation/database.html

9. On Windows, export the following registry key:

   ```
   HKEY_LOCAL_MACHINE\Software\oracle
   ```

   In addition, for system components, such as Oracle Web Cache, export the following Windows Registry key:

   ```
   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
   ```

   To export a key, use the following command:

   ```
   regedit /E  FileName Key
   ```

   For example:

   ```
   regedit /E C:\oracleregistry.reg HKEY_LOCAL_MACHINE/oracle
   ```

   You can also use the Registry Editor to export the key. See the Registry Editor Help for more information.

10. 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 on a regular basis and at the times described in Section 15.3.2.

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 3.4.2.

2. Back up the Administration Server domain directories. This backs up Java components such as Oracle SOA Suite and Oracle WebCenter. For example:

   ```
   tar -cf domain_home_backup_092909.tar MW_HOME/user_projects/domains/domain_
   name/*
   ```

   For Oracle Portal, Oracle Reports, Oracle Forms Services, and Oracle Business Intelligence Discoverer, you must back up the Managed Server domain directories, in addition to the Administration Server domain.

3. Back up the Oracle instance home. This backs up the system components, such as Oracle HTTP Server. For example:

   ```
   tar -cf sc_home_backup_092909.tar ORACLE_INSTANCE/*
   ```

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*, which is available at:

   http://www.oracle.com/technology/documentation/database.html

5. Create a record of your Oracle Fusion Middleware environment. See Section 16.4.

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 following information for each Oracle Fusion Middleware installation in your environment:
  - Installation type (for example, Oracle SOA Suite)
  - 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 Middleware home, Oracle Common home, Oracle homes, Oracle WebLogic Server domain home (if it does not reside in the user_projects directory in the Middleware home), and the Oracle instance home
  - Amount of disk space used by the installation
  - Port numbers used by the installation
- 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

17 Recovering Your Environment

This chapter describes recommended recovery strategies and procedures for recovering Oracle Fusion Middleware from different types of failures and outages.

This chapter includes the following topics:

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 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.

For example, for online recovery on Windows, use copy; for offline recovery on Windows, use copy, xcopy, or jar. Do not use Winzip because it does not work with long filenames or extensions.

For example, for Linux and UNIX, use tar.

Do not use the jar utility to archive and compress the file system. This avoids warnings or errors from the zip tool about zipping open files (for example, the *ORACLE_HOME*/jdk files).

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.

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 a Middleware Home
- Recovering an Oracle WebLogic Server Domain
- Recovering an Oracle Instance Home
- Recovering the Administration Server Configuration
- Recovering a Managed Server
- Recovering Components
- Recovering a Cluster
- Recovering Applications
- Recovering a Database

17.2.1 Recovering a Middleware Home

You can recover a Middleware home that was corrupted or from which files were deleted.

1. Stop all relevant processes. That is, stop all processes that are running from that Middleware home.

   For example, stop the Oracle WebLogic Server Administration Server processes and the Node Manager processes.

2. Recover the Middleware home directory from backup. For example:

```
cd MW_HOME
(UNIX) tar -xf mw_home_backup_092909.tar
(Windows) jar xtf mw_home_backup_092909.jar
```

3. Start all relevant processes. That is, start all processes that run in the Middleware home. For example, start the Administration Server:

```
DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
 -Dweblogic.management.password=password
 -Dweblogic.system.StoreBootIdentity=true
```

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. For example, stop the Administration Server and Managed Servers. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script to stop the Administration Server:

   ```
   DOMAIN_HOME/bin/stopWeblogic.sh username password admin_url
   ```

2. Recover the domain directory from backup:

   ```
   cd DOMAIN_HOME
   (UNIX) tar -xf domain_backup_092909.tar
   (Windows) jar xtf domain_backup_092909.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 -Dweblogic.management.username=username
    -Dweblogic.management.password=password
    -Dweblogic.system.StoreBootIdentity=true
   ```

17.2.3 Recovering an Oracle Instance Home

An Oracle instance home contains configuration information for system components, such as Oracle HTTP Server or Oracle Internet Directory. (See Section 3.5.2 for a list of system components.) The following topics describe how to recover an Oracle instance home:

- Recovering After Oracle Instance Home Deleted from File System
- Recovering After Oracle Instance Home Deregistered

17.2.3.1 Recovering After Oracle Instance Home Deleted from File System

To recover an Oracle instance home that was corrupted or deleted from the file system:

1. Stop all relevant processes. That is, kill all processes that are related to that Oracle instance
2. Recover the Oracle instance home directory from a backup file. For example:

   ```
   cd ORACLE_INSTANCE
   (UNIX) tar -xf sc_home_backup_092909.tar
   (Windows) jar xtf sc_home_backup_092909.jar
   ```

3. Start all relevant processes. That is, start all processes that are related to that Oracle instance:

   ```
   opmnctl startall
   ```

17.2.3.2 Recovering After Oracle Instance Home Deregistered

To recover an Oracle instance home that was deregistered from the domain:

1. Recover the Oracle instance home directory from a backup file. For example, on Linux:

   ```
   cd ORACLE_INSTANCE
   tar -xf Instance_home_backup_092909.tar
   ```

2. Register the Oracle instance, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command. For example:

```
opmnctl registerinstance -adminHost admin_server_host
     -adminPort admin_server_port -adminUsername username
     -adminPassword password
     -oracleInstance ORACLE_INSTANCE_dir -oracleHome ORACLE_HOME_dir
     -instanceName Instance_name -wlserverHome Middleware_Home
```

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. The Administration Server directory is regenerated automatically, except for security information. As a result, whenever you start the Administration Server, it prompts for a user name and password. To prevent this, you can recover the configuration.

> **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. You can use the Oracle WebLogic Server Administration Console, WLST, or a script. For example, to stop the Administration Server on Linux, use the following script:

```
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. You can use WLST or the following script:

```
DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
   -Dweblogic.management.password=password
   -Dweblogic.system.StoreBootIdentity=true
```

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

- Recovering an Oracle SOA Suite Managed Server That Has a Separate Directory

  This section pertains when Oracle SOA Suite is configured in a domain and no Managed Servers share the domain directory with the Administration Server.

17.2.5.1 Recovering a Managed Server When It Cannot Be Started

In this scenario, the Managed Server 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 or if the file system is lost, take the following steps:

   a. Recover the Middleware home from the backup, if required.

   ```
   tar -xf mw_home_backup_092909.tar
   ```

   b. Create a domain template jar file for the Administration Server, using the pack utility. For example:

   ```
   pack.sh -domain=MW_HOME/user_projects/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.

   c. Unpack the domain template jar file, using the unpack utility:

   ```
   unpack.sh -template=/scratch/aime1/ms.jar
      -domain=MW_HOME/user_projects/domains/domain_name
      -log=/scratch/logs/new.log -log_priority=info
   ```

   d. 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 takes care of pushing the bits to the stage directories in the Managed Server.
   - For no-stage and external-stage mode applications, ensure that application files are available in the stage directories of the Managed Server.

   See *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server* for information about stage, no-stage, and external-stage mode applications.

   e. Start the Managed Server. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script:

   ```
   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. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script:

```
DOMAIN_HOME/bin/stopManagedWeblogic.sh managed_server_name admin_url
        username password
```

2. Recover the Middleware home from the backup, if required:

```
tar -xf mw_home_backup_092909.tar
```

3. Create a domain template jar file for the Administration Server, using the pack utility. For example:

```
pack.sh -domain=MW_HOME/user_projects/domains/WLS_SOAWC
  -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.

4. Unpack the domain template jar file, using the unpack utility:

```
unpack.sh -template=/scratch/aime1/ms.jar
   -domain=MW_HOME/user_projects/domains/WLS_SOAWC
   -log=/scratch/logs/new.log -log_priority=info
```

5. 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 takes care of pushing the bits to the stage directories in the Managed Server.
- For no-stage and external-stage mode applications, ensure that application files are available in the stage directories of the Managed Server.

See *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

6. Restart the Managed Server. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name
          admin_url
```

17.2.5.3 Recovering an Oracle SOA Suite Managed Server That Has a Separate Directory

When Oracle SOA Suite is configured in a domain and no Managed Servers share the domain directory with the Administration Server, you must restore the Managed Server directory. For example, a domain contains two Managed Servers, one of which contains Oracle SOA Suite, but neither of the Managed Server's directories are in the same directory structure as the Administration Server.

In this case, you must restore the Managed Server from backup:

1. Restore the Managed Server from backup:

   ```
   cd ManagedServer_Home
   tar -xf managed_server_backup_092909.tar
   ```

2. Restart the Managed Server:

   ```
   DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name
             admin_url
   ```

17.2.6 Recovering Components

The following topics describe how to recover a component:

- Recovering After a Component's Files Are Deleted or Corrupted
- Recovering a Component That Is Not Functioning Properly After Configuration Change
- Recovering Components After Cluster Configuration Change

17.2.6.1 Recovering After a Component's Files Are Deleted or Corrupted

You can recover a component's files if they are deleted or corrupted. The steps you take depend on the type of component:

- For a Java component, such as Oracle SOA Suite, you recover the Managed Server, as described in Section 17.2.5.
- For system components, such as Oracle HTTP Server or Oracle Web Cache:

  1. Stop the component. For example, to stop Oracle HTTP Server:

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

     For information on stopping components, see Section 4.3.

  2. Recover the component-specific files from backup. Section 15.5 lists the directories and files needed for each component. For example, to recover Oracle HTTP Server files, you recover the following directories:

     ```
     ORACLE_INSTANCE/config/OHS/component_name
     ORACLE_INSTANCE/diagnostics/logs/OHS/component_name
     ```

  3. Start the component. For example, to start Oracle HTTP Server:

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

     For information on starting components, see Section 4.3.

17.2.6.2 Recovering a Component That Is Not Functioning Properly After Configuration Change

You can recover a component that 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.

The steps you take depend on the type of component:

- For a Java component, such as Oracle SOA Suite, you recover the Managed Server, as described in Section 17.2.5.
- For system components, such as Oracle HTTP Server:

  1. Stop the component. For example, to stop Oracle HTTP Server:

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

     For information on stopping components, see Section 4.3.

  2. Recover the component-specific files from backup. Section 15.5 lists the directories and files needed for each component. For example, to recover Oracle HTTP Server files, you recover the following directories:

     ```
     ORACLE_INSTANCE/config/OHS/component_name
     ORACLE_INSTANCE/diagnostics/logs/OHS/component_name
     ```

  3. Start the component. For example, to start Oracle HTTP Server:

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

     For information on starting components, see Section 4.3.

17.2.6.3 Recovering Components 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 all processes, such as the Managed Servers and the Administration Server. You can use the Oracle WebLogic Server Administration Console, WLST or a script. For example, to stop the Administration Server on Linux, use the following script:

   ```
   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. You can use WLST or the following script:

```
DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
 -Dweblogic.management.password=password
 -Dweblogic.system.StoreBootIdentity=true
```

4. Start the cluster. You can use the Oracle WebLogic Server Administration Console or WLST. For example, to use the WLST start command:

   ```
   start('clusterName', '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:

   ```
   stop('clusterName', 'Cluster')
   ```

2. Stop the Administration Server. You can use WLST or the following script:

   ```
   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. You can use WLST or the following script:

   ```
   DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
    -Dweblogic.management.password=password
    -Dweblogic.system.StoreBootIdentity=true
   ```

5. Start the cluster. You can use the Oracle WebLogic Server Administration Console or WLST. For example, to use WLST:

```
start('clusterName', '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. You can use the Oracle WebLogic Server Administration Console, WLST or a script. For example, to stop the Administration Server on Linux, use the following script:

   ```
   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. You can use WLST or the following script:

   ```
   DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
      -Dweblogic.management.password=password
      -Dweblogic.system.StoreBootIdentity=true
   ```

   The deleted member is now back in the cluster.

4. Start all processes, such as the Managed Servers. You can use WLST or a script. 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('clusterName', '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 no-stage 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:** *Oracle Fusion Middleware 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. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script:

   ```
   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 (such as SOA 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 Appendix D for the schemas used by each component.

For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*, which is available at:

http://www.oracle.com/technology/documentation/database.html

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 Administration Server Host
- Recovering After Loss of Managed Server Host
- Recovering After Loss of Component
- Additional Actions for Recovering Entities After Loss of Host
- Recovering After Loss of Host for a Database

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 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

17.3.1.1 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 needs to be recovered.

To recover the Administration Server:

1. Attempt to start the Administration Server. You can use WLST, or the following script:

   ```
   DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
    -Dweblogic.management.password=password
    -Dweblogic.system.StoreBootIdentity=true
   ```

   If the Administration Server starts, you do not need to take any further steps.

2. 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. For example, stop the Administration Server and Managed Servers. You can use WLST or the following script to stop the Administration Server:

      ```
      DOMAIN_HOME/bin/stopWeblogic.sh username password admin_url
      ```

   b. Recover the Middleware home, if needed:

      ```
      tar -xf mw_home_backup_092909.tar
      ```

   c. If the domain directory does not reside in the Middleware home, recover the domain directory from backup:

      ```
      cd DOMAIN_HOME
      tar -xf domain_backup_092909.tar
      ```

   d. Start the Administration Server. You can use WLST or the following script:

      ```
      DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
          -Dweblogic.management.password=password
          -Dweblogic.system.StoreBootIdentity=true
      ```

   e. Start the Node Manager:

      ```
      java weblogic.WLST
      wls:/offline> startNodeManager()
      ```

Now you can start and stop the Managed Server on Host B using the Administration Console running on Host A.

17.3.1.2 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 needs to be moved to Host C.

1. Recover the Middleware home to Host C (the new Host where the Administration Server will be recovered).

```
cd MW_HOME
tar -xf mw_home_backup_092909.tar
```

2. If the domain directory does not reside in the Middleware home, recover the domain directory from backup:

   ```
   cd DOMAIN_HOME
   tar -xf domain_backup_092909.tar
   ```

3. Start the Node Manager on Host C if it was configured on the original host:

   ```
   java weblogic.WLST
   wls:/offline> startNodeManager()
   ```

4. Start the Administration Server. You can use WLST or the following script:

   ```
   DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
    -Dweblogic.management.password=password
    -Dweblogic.system.StoreBootIdentity=true
   ```

5. Ensure that additional application artifacts are available. For example, if the deployment mode is no-stage 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.

6. Update Oracle Inventory, as described in Section 17.3.4.4.

7. Edit the targets.xml file for Fusion Middleware Control, as described in Section 17.3.4.2.

8. Oracle Management Service, which is part of Fusion Middleware Control, is on the original host and is recovered to the new host when you restore the Administration Server. Oracle Management Agent connects to Oracle Management Service to monitor certain components. If your environment contains components, such as Oracle Internet Directory and Oracle Virtual Directory, that use Oracle Management Agent, but they are located on a different host, you must take the following steps on each host containing the components. For example, the Administration Server was on Host A, but is restored, along with Oracle Management Service, to Host B. Oracle Internet Directory is on Host C and Oracle Virtual Directory is on Host D. You must take these steps on both Host C and Host D.

   a. Edit the following file:

      ```
      (UNIX) ORACLE_INSTANCE/EMAGENT/emagent_name/sysman/config/emd.properties
      (Windows)ORACLE_INSTANCE\EMAGENT\emagent_name\sysman\config\emd.properties
      ```

      Update the following entries, replacing the host name with the new host for the Administration Server:

      ```
      emdWalletSrcUrl=http://newhost.domain.com:port/em/wallets/emd
      REPOSITORY_URL=http://newhost.domain.com:port/em/upload/
      ```

   b. Shut down and restart the EM Agent process:

      ```
      cd $ORACLE_INSTANCE/EMAGENT/emagent_dir
      ./emctl stop agent
      ./emctl start agent
      ```

```
./emctl status agent
```

The status command will show the REPOSITORY_URL pointing to the new host.

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 17.3.4 for information about additional actions you must take.

17.3.2 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
- Recovering an Oracle SOA Suite Managed Server That Has a Separate Directory

  This section pertains when Oracle SOA Suite is configured in a domain and no Managed Servers share the domain directory with the Administration Server.

17.3.2.1 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 needs to be recovered to Host B.

1. Start the Node Manager on Host B:

   ```
   java weblogic.WLST
   wls:/offline> startNodeManager()
   ```

2. Start the Managed Server. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script:

   ```
   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. Stop the Node Manager:

      ```
       java weblogic.WLST
      wls:/offline> stopNodeManager()
      ```

   b. Recover the Middleware home to Host B from the backup, if required:

      ```
      tar -xf mw_home_backup_092909.tar
      ```

   c. If the Managed Server contains Oracle Portal, Oracle Reports, Oracle Forms Services, or Oracle Business Intelligence Discoverer, and the Managed Server domain directories reside outside of the Middleware home, restore the domain, in addition to the Middleware home. For example:

      ```
      cd Domain_Home
      ```

```
tar -xf domain_home_backup_092909.tar
```

Go to Step e.

d. If the Managed Server does not contain the components listed in Step c, take the following steps:

– Create a domain template jar file for the Administration Server running in Host A, using the pack utility. For example:

```
pack.sh -domain=MW_HOME/user_projects/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.

– Unpack the domain template jar file in Host B, using the unpack utility:

```
unpack.sh -template=/scratch/aime1/ms.jar
  -domain=MW_HOME/user_projects/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 no-stage or external stage mode, copy the application artifacts from 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 *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

f. If the Node Manager is not started, start it:

```
java weblogic.WLST
wls:/offline> startNodeManager()
```

g. Start the Managed Server. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

The Managed Server connects to the Administration Server and updates its configuration changes.

17.3.2.2 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 needs to be recovered to Host C.

> **Important:** Recover the Middleware home to the same location as the original.

Take the following steps:

1. Recover the MW_HOME for the Managed Server to Host C.

   ```
   tar -xf mw_home_backup_092909.tar
   ```

2. If the Managed Server contains Oracle Portal, Oracle Reports, Oracle Forms Services, or Oracle Business Intelligence Discoverer, and the Managed Server domain directories reside outside of the Middleware home, restore the domain, in addition to the Middleware home. For example:

   ```
   cd Domain_Home
   tar -xf domain_home_backup_092909.tar
   ```

   Go to Step 4.

3. If the Managed Server does not contain the components listed in Step 2, take the following steps:

   a. Create a domain template jar file from the Administration Server running in Host A, using the pack utility. For example:

   ```
   pack.sh -domain=MW_HOME/user_projects/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 on Host C, using the unpack utility:

   ```
   unpack.sh -template=/scratch/aime1/ms.jar
      -domain=MW_HOME/user_projects/domains/domain_name
      -log=/scratch/logs/new.log -log_priority=info
   ```

   If you are recovering to a different domain home, use the -app_dir switch in the unpack command.

4. 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 no-stage or external stage mode, copy the application artifacts from 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 *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

5. Start the Node Manager on Host C, if it is not started:

```
java weblogic.WLST
wls:/offline> startNodeManager()
```

6. Using WLST, connect to the Administration Server and then enroll the Node Manager running in the new host with the Administration Server:

```
connect('username','password','http://host:port')
nmEnroll('MW_HOME/user_projects/domains/domain_name',
  'MW_HOME/wlserver_n/common/nodemanager')
```

7. Change the Managed Server configuration to point to the new host:

   a. In the WebLogic Server Administration Console, 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.

      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 *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

   b. Change the Managed Server configuration to point to the new machine. (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. In the **Machine** field, select the machine to which you want to assign the server.)

      Change **Listen Address** to the new host. (If the listening address was set to blank, you do not need to change it.)

8. Start the Managed Server. You can use the Oracle WebLogic Server Administration Console, WLST, or the following script:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name
          admin_url
```

   The Managed Server connects to the Administration Server and updates its configuration changes.

9. Update Oracle Inventory, as described in Section 17.3.4.4.

10. Edit the targets.xml file for Fusion Middleware Control, as described in Section 17.3.4.2.

Now you can start and stop the Managed Server on Host C using the Administration Server running on Host A.

17.3.2.3 Recovering an Oracle SOA Suite Managed Server That Has a Separate Directory

When Oracle SOA Suite is configured in a domain and no Managed Servers share the domain directory with the Administration Server, you must restore the Managed Server directory. For example, a domain contains two Managed Servers, one of which contains Oracle SOA Suite, but neither of the Managed Server's directories are in the same directory structure as the Administration Server.

You use the following steps when you are restoring to the same host or a different host:

1. Restore the Managed Server from backup:

```
cd ManagedServer_Home
```

```
tar -xf managed_server_backup_092909.tar
```

2. Restart the Managed Server:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

17.3.3 Recovering After Loss of Component

If you lose a host that contains a component (and its Managed Server, if applicable), you can recover it to the same host or a different host, as 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

17.3.3.1 Recovering a Java Component to the Same Host

To recover a Java component, such as Oracle SOA Suite:

1. Recover the Managed Server, as described in Section 17.2.5.1.

However, note that some components require additional steps, which are described in subsequent sections.

17.3.3.2 Recovering a Java Component to a Different Host

To recover a Java component, such as Oracle SOA Suite:

1. Recover the Managed Server, as described in Section 17.3.2.2.
2. Edit the targets.xml file for Fusion Middleware Control, as described in Section 17.3.4.2.

However, note that some components require additional steps, which are described in subsequent sections.

17.3.3.3 Recovering a System Component to the Same Host

To recover a system component, such as Oracle HTTP Server, you take the following general steps. However, note that some components require additional steps.

1. Stop all relevant processes. That is, stop all processes that are related to the component. For example, to stop Oracle HTTP Server:

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

   For information on stopping components, see Section 4.3.

2. Recover the component-specific files from backup. Section 15.5 lists the directories and files needed for each component. For example, to recover Oracle HTTP Server files, you recover the following directories:

   ```
   ORACLE_INSTANCE/config/OHS/component_name
   ORACLE_INSTANCE/diagnostics/logs/OHS/component_name
   ```

3. If the Oracle instance home has been deregistered from the Administration Server, register the Oracle instance:

   ```
   opmnctl registerinstance -adminHost admin_server_host
        -adminPort admin_server_port -adminUsername username
   ```

```
     -adminPassword password
     -wlserverHome wlserver_home_location
```

If only the file system is being recovered, you do not need to register the Oracle instance.

4. Start all relevant processes, as described in Section 4.3.

17.3.3.4 Recovering a System Component to a Different Host

To recover a system component, such as Oracle HTTP Server, you take the following general steps. However, note that most components require additional steps, which are described in separate sections.

1. Recover the Middleware home, as described in Section 17.2.1.
2. Start all relevant processes. Section 4.3 explains how to start components.
3. Update the registration of the Oracle instance with the Administration Server, using the opmnctl `updateinstanceregistration` command on the new host. For example:

   ```
   opmnctl updateinstanceregistration -adminHost admin_server_host
   ```

   This command updates OPMN's instance.properties file.
4. Update the registration of the component with the Administration Server, using the opmnctl `updatecomponentregistration` command on the new host. For example, to update the registration for Oracle Virtual Directory, use the following command:

   ```
   opmnctl updatecomponentregistration -Host new_host -Port nonSSLPort
       -componentName ovd1 -componentType OVD
   ```
5. Edit the targets.xml file for Fusion Middleware Control, as described in Section 17.3.4.2.

Depending on the components, you may need to take additional actions. See the subsequent sections for more information.

17.3.3.5 Recovering Oracle SOA Suite After Loss of Host

Note that when Oracle SOA Suite is configured in a domain and no Managed Servers share the domain directory with the Administration Server, take the steps described in Section 17.3.2.3. Otherwise, follow the steps in this section.

To recover the Oracle SOA Suite Managed Server to the same host after loss of host:

1. Recover the Managed Server, as described in Section 17.3.2.1.

To recover the Oracle SOA Suite Managed Server to a different host after loss of host:

1. Before you recover, update the WSDL file to point to the new host name and port.
2. Recover the Managed Server, as described in Section 17.3.2.2.
3. After you recover the Oracle SOA Suite Managed Server, take the following actions:
   - If the ant command is used to deploy composites, edit the deploy-sar.xml file, which is located in:

     ```
     (UNIX) ORACLE_HOME/bin
     (Windows) ORACLE_HOME\bin
     ```

In the following line, replace the previous host name with the new host name:

```
<property name="wlsHost" value="newhostname"/>
```

If a Load Balancer is used, do not modify this property. Instead, register the new host with the Load Balancer.

- Change the host name in the soa-infra MBean:

  a. In Fusion Middleware Control, navigate to the Managed Server.

  b. From the WebLogic Server menu, choose **System MBean Browser.**

  c. Expand **Application Defined MBeans,** then **oracle.as.soainfra.config**, then **Server:** ***server_name*** and then **SoaInfraConfig.** Select **soa-infra.**

  d. In the Attributes tab, click **ServerURL.** If the ServerURL attribute contains a value, change the host name to the new host name.

  e. Click **Apply.**

- Redeploy all applications which have the WSDL files updated to the new host name.

Note: If there is no Load Balancer configured with the environment and Oracle SOA Suite needs to be recovered to a different host, then in-flight instances that are pending a response from task flow and asynchronous responses are not recovered. Oracle recommends that you use a Load Balancer to ensure that you can recover to a different host.

If a Load Balancer is configured with the environment, follow steps in Section 17.3.2.2. Then, take the following additional steps:

1. Log in to the Oracle WebLogic Server Administration Server.
2. In Domain Structure, navigate to Servers. For each Managed Server, select the Protocol tab, then the HTTP tab.
3. For **Frontend Host,** enter the new host name.
4. For **Frontend HTTP Port** and **Frontend HTTPs Port,** if applicable, enter the new port number.
5. Restart each Managed Server.

17.3.3.6 Recovering Oracle Business Activity Monitoring to a Different Host

To recover Oracle Business Activity Monitoring to a different host:

1. Recover the Managed Server, as described in Section 17.3.2.2.

17.3.3.7 Recovering Oracle WebCenter to a Different Host

To recover Oracle WebCenter to a different host:

1. Recover the Managed Server, as described in Section 17.3.2.2.

17.3.3.8 Recovering Web Tier Components to a Different Host

The Web tier consists of Oracle HTTP Server and Oracle Web Cache. The following topics describe how to recover these components:

- Recovering Oracle HTTP Server to a Different Host
- Recovering Oracle Web Cache to a Different Host

17.3.3.8.1 Recovering Oracle HTTP Server to a Different Host To recover Oracle HTTP Server to a different host:

1. Recover the component as described in Section 17.3.3.4.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.
3. Modify the ServerName entry in the following file to have the new host name:

   ```
   (UNIX) ORACLE_INSTANCE/config/OHS/ohs_name/httpd.conf
   (Windows) ORACLE_INSTANCE\config\OHS\ohs_name\httpd.conf
   ```

17.3.3.8.2 Recovering Oracle Web Cache to a Different Host To recover Oracle Web Cache to a different host:

1. Recover the component as described in Section 17.3.3.4.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.
3. Edit the webcache.xml file, replacing the previous host name with the new host name. The file is located in:

   ```
   (UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
   (Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
   ```

17.3.3.9 Recovering Identity Management Components to a Different Host

For most Identity Management components, you recover the Managed Server, as described in Section 17.3.2.2.

Some components require additional steps to recover the components to a different host, as described in the following topics:

- Recovering Oracle Internet Directory to a Different Host
- Recovering Oracle Virtual Directory to a Different Host
- Recovering Oracle Directory Integration Platform to a Different Host
- Recovering Oracle Directory Services Manager to a Different Host
- Recovering Oracle Identity Federation to a Different Host

17.3.3.9.1 Recovering Oracle Internet Directory to a Different Host To recover Oracle Internet Directory to a different host:

1. Recover the component as described in Section 17.3.3.4.
2. On UNIX and Linux systems, before you attempt to start Oracle Internet Directory, set the following to have root permission:

   ```
   ORACLE_HOME/bin/oidldapd
   ```

   For example:

   ```
   chown root oidldapd
   chmod 4710 oidldapd
   ```

3. Recover Oracle Management Agent, as described in Section 17.3.4.3.

4. If the Managed Server on which Oracle Directory Services Manager is deployed is moved to different host and if SSL is enabled, you must delete the following file on the new host:

   ```
   DOMAIN_HOME/servers/wls_ods1/tmp/_WL_user/odsm_
   11.1.1.1.0/randomid/war/conf/odsm.cer
   ```

   Oracle Directory Services Manager uses this file as its key store and trust store and the password is stored in JKS. However, when Oracle Directory Services Manager is copied to another host and is started, it generates a different password than odsm.cer. If you delete the file, Oracle Directory Services Manager creates a new file when it starts.

17.3.3.9.2 Recovering Oracle Virtual Directory to a Different Host To recover Oracle Virtual Directory to a different host:

1. Recover the component as described in Section 17.3.3.4.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.

17.3.3.9.3 Recovering Oracle Directory Integration Platform to a Different Host To recover Oracle Directory Integration Platform to a different host:

1. Recover the Managed Server, as described in Section 17.3.2.2.
2. Before starting the Managed Server, restore the files in the following directory:

   ```
   DOMAIN_HOME/servers/wls_ods1/stage/DIP/11.1.1.1.0/
   ```

3. Start the Managed Servers and Oracle instances.
4. If Oracle Internet Directory is also moved to a different host, execute the following commands immediately after the Managed Server and the Oracle instance are started:

   ```
   set ORACLE_HOME Oracle_home_path
   set WLS_HOME WLS_Home_path
   cd ORACLE_HOME/bin
   ./manageDIPServerConfig set -h dip_server_host -p dip_server_port
       -D weblogic_user -attribute oidhostport -value oid_host:oid_ssl_port
   ```

   The manageDIPServerConfig command prompts you for a password.

   For example:

   ```
   ./manageDIPServerConfig set -h hostname -p 19523 -D weblogic
               -attribute oidhostport -value hostname.domain.com:24163
   ```

5. Register the Oracle instance, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command on the new host. For example:

   ```
   opmnctl registerinstance -adminHost admin_server_host
        -adminPort admin_server_port -adminUsername username
        -adminPassword password
        -wlserverHome wlserver_home_location
   ```

17.3.3.9.4 Recovering Oracle Directory Services Manager to a Different Host To recover Oracle Directory Services Manager to a different host:

1. Recover the component as described in Section 17.3.3.4.

17.3.3.9.5 Recovering Oracle Identity Federation to a Different Host Because Oracle Identity Federation provides SSO functionality, if the host name on which Oracle Identity Federation runs is changed as part of loss of host recovery, it impacts remote partners. In that case, remote partners must make changes regarding the host name to continue to operate. It may take many days for remote partners to update their data and this will cause production delays that are unacceptable. Oracle strongly recommends that you do not change the host name of a standalone Oracle Identity Federation server.

If a load balancer is part of the environment and the host where Oracle Identity Federation is being recovered is in the list of VIPs, then no host name changes are required.

In the case of a standalone installation of Oracle Identity Federation, Oracle recommends using a new host with the same name to minimize the impact. However, if, for whatever reason, you need to use a different host name for recovering Oracle Identity Federation, then the host name needs to be updated manually for Oracle Identity Federation and remote partners.

To recover Oracle Identity Federation to a different host:

1. Recover the Managed Server, as described in Section 17.3.2.2.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.
3. Register the Oracle instance, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command on the new host. For example:

   ```
   opmnctl registerinstance -adminHost admin_server_host
        -adminPort admin_server_port -adminUsername username
        -adminPassword password
        -wlserverHome wlserver_home_location
   ```

4. Provide the updated data to remote partners.
5. Modify the host name using Fusion Middleware Control:
   a. In the navigation pane, expand the farm and then **Identity and Access.**
   b. Select the Oracle Identity Federation instance.
   c. From the Oracle Identity Federation menu, choose **Administration,** then **Server Properties.**

      The Server Properties page is displayed.
   d. For **Host,** replace the old host name with the new host name.
   e. For **Port,** replace the port number if it has changed.
   f. For **SOAP Port,** replace the port number if it has changed.
   g. Click **Apply.**
   h. Restart the Managed Server to which Oracle Identity Federation is deployed:

      ```
      DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name
               admin_url
      ```

6. If Oracle Identity Federation is acting as an SSL server, you must replace the SSL certificate presented by Oracle Identity Federation to clients with a new one that has the new host name. Otherwise, host name verification by clients may fail.

17.3.3.10 Recovering Oracle Portal, Oracle Reports, Oracle Forms Services, and Oracle Business Intelligence Discoverer to a Different Host

The following topics describe how to recover these components to a different host:

- Recovering Oracle Portal to a Different Host
- Recovering Oracle Business Intelligence Discoverer to a Different Host
- Recovering Oracle Reports to a Different Host
- Recovering Oracle Forms Services to a Different Host

17.3.3.10.1 Recovering Oracle Portal to a Different Host To recover Oracle Portal to a different host:

1. Restore the Middleware home, domain directory, and the Oracle instance directory to the new host. See Section 17.3.2.2 for more information.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.
3. Register the Oracle instance, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command on the new host. For example:

   ```
   opmnctl registerinstance -adminHost admin_server_host
        -adminPort admin_server_port -adminUsername username
        -adminPassword password
        -wlserverHome wlserver_home_location
   ```

4. Modify the following files, replacing the old host name with the new host name:

   ```
   ORACLE_INSTANCE/config/OHS/ohs_name/httpd.conf
   ORACLE_INSTANCE/config/OHS/ohs_name/moduleconf/portal.conf
   ```

5. Run the ssoreg script, which is located in:

   ```
   Identity_Management_ORACLE_HOME/sso/bin
   ```

   Use the following command:

   ```
   ssoreg.sh -site_name newhost:http_listen_port
       -mod_osso_url http://newhost:http_listen_port -config_mod_osso TRUE
       -oracle_home_path $ORACLE_HOME -config_file any_new_file_path
       -admin_info cn=orcladmin -virtualhost -remote_midtier
   ```

   For example:

   ```
   ssoreg.sh -site_name example.com:8090
      -mod_osso_url http://example.com:8090 -config_mod_osso TRUE
      -oracle_home_path $ORACLE_HOME -config_file /tmp/loh_osso.conf
      -admin_info cn=orcladmin -virtualhost -remote_midtier
   ```

6. Copy the file from the previous step to the new host.
7. In the new host, modify the OssoConfigFile section in the following file to include the path of the file in step 5:

   ```
   ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/mod_osso.conf
   ```

   For example:

   ```
   <IfModule mod_osso.c>
       OssoIpCheck off
       OssoSecureCookies off
   ```

```
OssoIdleTimeout off
OssoConfigFile /tmp/path_of_file_created
```

8. Edit the following files, replacing the previous host name with the new host name:

   - webcache.xml. This file is located in:

     ```
     (UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
     (Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
     ```

     Replace all occurrences of the previous host name with the new host name.

   - instance.properties. The file is located in:

     ```
     (UNIX) ORACLE_INSTANCE/config/OPMN/opmn
     (Windows) ORACLE_INSTANCE\config\OPMN\opmn
     ```

     In the following line, replace the previous host name with the new host name if the Administration Server host name has changed.

     ```
     adminHost=host_name
     ```

9. If the published host used to access Oracle Portal is changing, take the following steps. This could happen if you have a single node install which contains both Oracle Web Cache and WLS_PORTAL, and those processes need to move to a different host. Another scenario is when you have Oracle Web Cache running on a node remotely from WLS_PORTAL, and Oracle Web Cache needs to move to a different host. In both these cases, take the following steps to update the Published Host information within Oracle Portal. (Note: If you have a load balancer or reverse proxy configuration, the steps are not needed.)

   a. Recursively delete all content from the following directory, but do not delete the directory itself:

      ```
      ORACLE_INSTANCE/portal/cache
      ```

   b. Log in to Fusion Middleware Control. Expand the farm and right-click **Portal.** Then, choose **Settings,** then **Wire Configuration.**

   c. In the Portal Midtier section, update **Published Host** with the new host name.

   d. In the Oracle Web Cache section, update **Host** with the new host name.

10. Restart the WLS_PORTAL instance.

17.3.3.10.2 Recovering Oracle Business Intelligence Discoverer to a Different Host To recover Oracle Business Intelligence Discoverer to a different host:

1. Recover the Managed Server as described in Section 17.3.2.2.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.
3. Register the Oracle instance, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command on the new host. For example:

   ```
   opmnctl registerinstance -adminHost admin_server_host
        -adminPort admin_server_port -adminUsername username
        -adminPassword password
        -wlserverHome wlserver_home_location
   ```

4. Edit the following files, replacing the previous host name with the new host name:

   - module_disco.conf. This file is located in:

```
(UNIX) ORACLE_INSTANCE/config/OHS/ohs_name//moduleconf
(Windows) ORACLE_INSTANCE\config\OHS\ohs_name\moduleconf
```

- webcache.xml. This file is located in:

  ```
  (UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
  (Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
  ```

  Replace all occurrences of the previous host name with the new host name.

5. Run the ssoreg script, which is located in:

   ```
   Identity_Management_ORACLE_HOME/sso/bin
   ```

   Use the following command:

   ```
   ssoreg.sh -site_name newhost:http_listen_port
       -mod_osso_url http://newhost:http_listen_port -config_mod_osso TRUE
       -oracle_home_path $ORACLE_HOME -config_file any_new_file_path
       -admin_info cn=orcladmin -virtualhost -remote_midtier
   ```

   For example:

   ```
   ssoreg.sh -site_name example.com:8090
      -mod_osso_url http://example.com:8090 -config_mod_osso TRUE
      -oracle_home_path $ORACLE_HOME -config_file /tmp/loh_osso.conf
      -admin_info cn=orcladmin -virtualhost -remote_midtier
   ```

6. Copy the file from the previous step to the new host.
7. In the new host, modify the OssoConfigFile section in the following file to include the path of the file in step 5:

   ```
   ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/mod_osso.conf
   ```

   For example:

   ```
   <IfModule mod_osso.c>
       OssoIpCheck off
       OssoSecureCookies off
       OssoIdleTimeout off
       OssoConfigFile /tmp/path_of_file_created
   ```

17.3.3.10.3 Recovering Oracle Reports to a Different Host To recover Oracle Reports to a different host:

1. Recover the Managed Server as described in Section 17.3.2.2.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.
3. Register the Oracle instance, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command on the new host. For example:

   ```
   opmnctl registerinstance -adminHost admin_server_host
        -adminPort admin_server_port -adminUsername username
        -adminPassword password
        -wlserverHome wlserver_home_location
   ```

4. Edit the following files, replacing the previous host name with the new host name:

   – reports_install.properties. The file is located in:

   ```
   (UNIX) ORACLE_INSTANCE/reports
   (Windows) ORACLE_INSTANCE\reports
   ```

Edit the parameters SERVER_NAME, OHS_HOST and REPORTS_MANAGED_WLS_HOST.

- webcache.xml. This file is located in:

  ```
  (UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
  (Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
  ```

  Replace all occurrences of the previous host name with the new host name.

- instance.properties. The file is located in:

  ```
  (UNIX) ORACLE_INSTANCE/config/OPMN/opmn
  (Windows) ORACLE_INSTANCE\config\OPMN\opmn
  ```

  In the following line, replace the previous host name with the new host name if the Administration Server host name has changed.

  ```
  adminHost=host_name
  ```

- reports_ohs.conf. The file is located in:

  ```
  (UNIX) ORACLE_INSTANCE/config/OHS/ohs_name/moduleconf
  (Windows) ORACLE_INSTANCE\config\OHS\ohs_name\moduleconf
  ```

- rwservlet.properties. The file is located in:

  ```
  (UNIX) MW_HOME/user_projects/domains/domain_name/servers/WLS_
  REPORTS/stage/reports/reports/configuration
  (Windows) MW_HOME\user_projects\domains\domain_name\servers/WLS_
  REPORTS\stage\reports\reports\configuration
  ```

  In the file, modify the <server> element to use the new host name.

5. In the following directory, rename the subdirectory to have the new host name:

   ```
   (UNIX) ORACLE_INSTANCE/diagnostics/logs/ReportsServer
   (Windows) ORACLE_INSTANCE\diagnostics\logs\ReportsServer
   ```

6. In the following directory, rename the *old_host_name*.dat file to the new host name:

   ```
   (UNIX) ORACLE_INSTANCE/reports/server
   (Windows) ORACLE_INSTANCE\reports\server
   ```

7. In the following directory, rename the subdirectory to have the new host name:

   ```
   (UNIX) ORACLE_INSTANCE/config/ReportsServer
   (Windows) ORACLE_INSTANCE\config\ReportsServer
   ```

8. Run the ssoreg script, which is located in:

   ```
   Identity_Management_ORACLE_HOME/sso/bin
   ```

   Use the following command:

   ```
   ssoreg.sh -site_name newhost:http_listen_port
       -mod_osso_url http://newhost:http_listen_port -config_mod_osso TRUE
       -oracle_home_path $ORACLE_HOME -config_file any_new_file_path
       -admin_info cn=orcladmin -virtualhost -remote_midtier
   ```

   For example:

   ```
   ssoreg.sh -site_name example.com:8090
      -mod_osso_url http://example.com:8090 -config_mod_osso TRUE
   ```

```
-oracle_home_path $ORACLE_HOME -config_file /tmp/loh_osso.conf
-admin_info cn=orcladmin -virtualhost -remote_midtier
```

9. Copy the file from the previous step to the new host.
10. In the new host, modify the OssoConfigFile section in the following file to include the path of the file in step 8:

    ```
    ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/mod_osso.conf
    ```

    For example:

    ```
    <IfModule mod_osso.c>
        OssoIpCheck off
        OssoSecureCookies off
        OssoIdleTimeout off
        OssoConfigFile /tmp/path_of_file_created
    ```

17.3.3.10.4 Recovering Oracle Forms Services to a Different Host To recover Oracle Forms Services to a different host:

1. Recover the Managed Server as described in Section 17.3.2.2.
2. Recover Oracle Management Agent, as described in Section 17.3.4.3.
3. Register the Oracle instance, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command on the new host. For example:

    ```
    opmnctl registerinstance -adminHost admin_server_host
         -adminPort admin_server_port -adminUsername username
         -adminPassword password
         -wlserverHome wlserver_home_location
    ```

4. Edit the following files, replacing the previous host name with the new host name:
    - webcache.xml. This file is located in:

      ```
      (UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
      (Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
      ```

      Replace all occurrences of the previous host name with the new host name.
    - instance.properties. The file is located in:

      ```
      (UNIX) ORACLE_INSTANCE/config/OPMN/opmn
      (Windows) ORACLE_INSTANCE\config\OPMN\opmn
      ```

      In the following line, replace the previous host name with the new host name if the Administration Server host name has changed.

      ```
      adminHost=host_name
      ```

    - forms.conf. The file is located in:

      ```
      (UNIX) ORACLE_INSTANCE/config/OHS/ohs_name/moduleconf
      (Windows) ORACLE_INSTANCE\config\OHS\ohs_name\moduleconf
      ```

      Replace the host name in the parameter WebLogicHost with the name of the new host.
5. On the Administration Server host, edit the following file:

    ```
    DOMAIN_HOME/opmn/topology.xml
    ```

Add properties for the <ias-component id> element for Oracle Forms Services. The following example shows the element after you modify it:

```
</ias-component>
  <ias-component id="forms" type="FormsComponent" >
  <em-properties>
      <property name="OracleHome" value="/path_to_oracle_home" />
      <property name="instName" value="instance_name" />
      <property name="EMTargetType" value="oracle_forms" />
      <property name="version" value="11.1.1" />
  </em-properties>
 </ias-component>
```

6. On the host where the Oracle instance has been recovered, update the registration of the component with the Administration Server, using the opmnctl `updatecomponentregistration` command on the new host.

   For example:

   ```
   opmnctl updatecomponentregistration -Host new_host -Port nonSSLPort
       -componentName forms -componentType FormsComponent
   ```

7. Run the ssoreg script, which is located in:

   ```
   Identity_Management_ORACLE_HOME/sso/bin
   ```

   Use the following command:

   ```
   ssoreg.sh -site_name newhost:http_listen_port
       -mod_osso_url http://newhost:http_listen_port -config_mod_osso TRUE
       -oracle_home_path $ORACLE_HOME -config_file any_new_file_path
       -admin_info cn=orcladmin -virtualhost -remote_midtier
   ```

   For example:

   ```
   ssoreg.sh -site_name example.com:8090
      -mod_osso_url http://example.com:8090 -config_mod_osso TRUE
      -oracle_home_path $ORACLE_HOME -config_file /tmp/loh_osso.conf
      -admin_info cn=orcladmin -virtualhost -remote_midtier
   ```

8. Copy the file from the previous step to the new host.

9. In the new host, modify the OssoConfigFile section in the following file to include the path of the file in step 7:

   ```
   ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/mod_osso.conf
   ```

   For example:

   ```
   <IfModule mod_osso.c>
       OssoIpCheck off
       OssoSecureCookies off
       OssoIdleTimeout off
       OssoConfigFile /tmp/path_of_file_created
   ```

17.3.4 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.

- Recovering Fusion Middleware Control to a Different Host

- Changing the Host Name in the targets.xml File for Fusion Middleware Control
- Recovering Oracle Management Agent When Components Are Recovered to a Different Host
- Updating Oracle Inventory
- Recover the Windows Registry

17.3.4.1 Recovering Fusion Middleware Control to a Different Host

To recover Fusion Middleware Control to a different host, take the following steps:

1. Update the host name in the following file:

   ```
   MW_HOME/user_projects/domains/domain_name/servers
   /AdminServer/tmp/_WL_user/em/hsz5x1/META-INF/emoms.properties
   ```

   In the file, change the host name for the following properties:

   ```
   mas.conn.url
   oracle.sysman.emSDK.svlt.ConsoleServerHost
   ```

2. Edit the following file:

   ```
   (UNIX) ORACLE_INSTANCE/EMAGENT/emagent_name/sysman/config/emd.properties
   (Windows) ORACLE_INSTANCE\EMAGENT\emagent_name\sysman\config\emd.properties
   ```

   In the file, edit the following entry for each component monitored by Oracle Management Agent, replacing the host name:

   ```
   REPOSITORY_URL=http://newhost.domain.com:port/em/upload/
   ```

17.3.4.2 Changing the Host Name in the targets.xml File for Fusion Middleware Control

When you recover a component to a different host, you must update the targets.xml file for Fusion Middleware Control. The file is located at:

```
MW_HOME/user_projects/domains/domain_name/sysman/state/targets.xml
```

In the file, change the host name to the new host name for components that are recovered to a different host.

17.3.4.3 Recovering Oracle Management Agent When Components Are Recovered to a Different Host

For many components, when you recover to a different host, as in the case of loss of host, you must take actions to recover Oracle Management Agent so that Fusion Middleware Control can manage the components. This pertains to the following installation types and components:

- Identity Management components
- Oracle Identity Federation
- Oracle Portal
- Oracle Business Intelligence Discoverer
- Oracle Forms Services
- Oracle Reports

To recover Oracle Management Agent, take the following actions:

1. Edit the following file:

   ```
   (UNIX) ORACLE_INSTANCE/EMAGENT/emagent_name/sysman/emd/targets.xml
   (Windows) ORACLE_INSTANCE\EMAGENT\emagent_name\sysman\emd\targets.xml
   ```

   In the file, edit the following element, replacing the host name:

   ```
   <Target TYPE="host" NAME="newhost.domain.com"
        DISPLAY_NAME="newhost.domain.com"/>
   ```

2. Edit the following file:

   ```
   (UNIX) ORACLE_INSTANCE/EMAGENT/emagent_name/sysman/config/emd.properties
   (Windows) ORACLE_INSTANCE\EMAGENT\emagent_name\sysman\config\emd.properties
   ```

   Update the following entry, replacing the host name:

   ```
   EMD_URL=http://newhost.domain.com:port/emd/main
   ```

3. Start Oracle Management Agent, using the following command:

   ```
   opmnctl startproc ias-component=EMAGENT
   ```

4. Start the Administration Server:

   ```
   DOMAIN_HOME/bin/startWebLogic.sh -Dweblogic.management.username=username
    -Dweblogic.management.password=password
    -Dweblogic.system.StoreBootIdentity=true
   ```

   Starting the Administration Server also starts Fusion Middleware Control.

17.3.4.4 Updating Oracle Inventory

For many components, when you recover to a different host, as in the case of loss of host, you need to update the Oracle inventory. To do so, execute the following script:

```
ORACLE_HOME/oui/bin/attachHome.sh
```

In addition, you must update beahomelist to edit the location of a Middleware home. Edit the following file to update the Middleware home information:

```
(UNIX) user_home/bea/beahomelist
(Windows) C:\bea\beahomelist
```

17.3.4.5 Recover the Windows Registry

When you recover any component to a different host on Windows, as in the case of loss of host, you need to recover the following Windows Registry key.

```
HKEY_LOCAL_MACHINE\Software\oracle
```

In addition, when you recover system components, such as Oracle Web Cache, you must recover the following Windows Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\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.

17.3.5 Recovering After Loss of Host for a Database

If the host that contained your database is lost, you can recover it using RMAN. You can recover the database at the desired granularity, either a full recovery or a tablespace recovery.

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. Note the following:

- See Appendix D for the schemas used by each component.
- For Oracle BPEL Process Manager, point-in-time recovery ensures that the latest process definitions and in-flight instances are restored. However, this may result in reexecution of the process steps. Oracle recommends that you strive for idempotent Oracle BPEL Process Manager processes. If the system contains processes that are not idempotent, you must clean them up from the dehydration store before starting Oracle Fusion Middleware. See *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite* for more information.

For detailed steps, see the *Oracle Database Backup and Recovery User's Guide,* which is available at:

http://www.oracle.com/technology/documentation/database.html

Part VIII

Advanced Administration: Expanding Your Environment

This part describes how to copy all or part of your Oracle Fusion Middleware environment.

It contains the following chapters:

18 Scaling Your Environment

You can expand your environment by adding Managed Servers, expanding your domain to include other products, creating a cluster of Managed Servers, or cloning existing Middleware homes and Oracle homes, as described by the following topics:

- Overview of Scaling Your Environment
- Extending a Domain to Support Additional Components
- Adding Additional Managed Servers to a Domain
- Creating Clusters
- Cloning a Middleware Home or Component

18.1 Overview of Scaling 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. Horizontally, Oracle Fusion Middleware can increase its throughput with several Managed Servers grouped together to share a workload. Also, Oracle Fusion Middleware provides great vertical scalability, allowing you to add more Managed Servers or components in the same host.

High availability refers to the ability of users to access a system. Deploying a high availability system minimizes the time when the system is down, or unavailable and maximizes the time when it is running, or 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:** *Oracle Fusion Middleware High Availability Guide* for more information about high availability

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 the Oracle SOA Suite. If you want to add other components, such as Oracle WebCenter, 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.

Table 18–1 shows which components you can add to an existing domain and the domain templates needed.

Table 18–1 Supported Domain Extensions

Existing Domain Template	Components That Can Be Added
Oracle SOA Suite	Any Oracle SOA Suite component. Any Oracle WebCenter component. Extend with Oracle WebCenter domain template. Any Web Tier component. Extend with Web Tier domain template.
Oracle Identity Management	Any Identity Management component. Any Web Tier component. Extend with Web Tier domain template.
Oracle Portal, Oracle Reports, Oracle Forms Services, Oracle Business Intelligence Discoverer	Any of these components. Any Web Tier component. Extend with Web Tier domain template.

> **Note:** For Identity Management components, Oracle Portal, Oracle Forms Services, Oracle Reports, and Oracle Business Intelligence Discoverer, if the component that you want to include in the domain is on a remote host, the WebLogic Server home must have the same full path as the WebLogic Server home for the original component.
>
> For example, you are extending a domain that initially was created to support Oracle SOA Suite so that it can now also support Oracle WebCenter and you install Oracle WebCenter on a remote host. In this case, if the WebLogic Server home for Oracle SOA Suite is located in /scratch/oracle/Middleware/wlserver_10.3, the WebLogic Server home for Oracle WebCenter must also be located in /scratch/oracle/Middleware/wlserver_10.3.

For example, to extend a domain that initially was created to support Oracle SOA Suite so that it can now also support Oracle WebCenter:

1. Use RCU to add any required schemas for the component, as described in *Oracle Fusion Middleware Repository Creation Utility User's Guide*.
2. Install Oracle WebCenter, as described in the *Oracle Fusion Middleware Installation Guide for Oracle WebCenter*.

3. From an Oracle home that was installed for the component you want to add (for example, for Oracle WebCenter), invoke the Configuration Wizard, using the following command:

   ```
   (UNIX) ORACLE_HOME/common/bin/config.sh
   (Windows) ORACLE_HOME\common\bin\config.cmd
   ```

   The Configuration Wizard's Welcome screen is displayed.

4. Select **Extend an existing WebLogic Domain.**
5. Click **Next.**

   The Select a WebLogic Domain Directory screen is displayed.

6. Select the directory for the domain to which you want to add the components.
7. Click **Next.**

   The Select Extension Source screen is displayed.

8. Select the source from which this domain will be extended. For example, select **Oracle WebCenter Spaces.**
9. Click **Next.**

   The Conflict Detected dialog box is displayed.

10. Select **Keep existing component** and **Apply this selection if further conflicts are detected.** Click **OK.**

    The Configure JDBC Data Sources screen is displayed.

11. Enter 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.

12. Click **Next.**

    The Customize Server and Cluster Configuration screen is displayed.

13. In this and the following customization screens, you can choose to customize. To do so, click **Yes.** If you do not want to customize the settings, click **No.**
14. Click **Next.**
15. Click **Next.**

The Review WebLogic Domain screen is displayed.

16. Review the information on the screen and if it is correct, click **Next.**

    The Extend WebLogic Domain screen is displayed.

17. Click **Extend.**
18. 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 Server 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 added to a domain, it does not automatically inherit the applications and services from the template.

To add a Managed Server to a domain, you can use the Oracle WebLogic Server Administration Console or WLST.

> **See:** Administration Console Online Help and *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for complete information about adding Managed Servers.

To add a Managed Server to a domain using Oracle WebLogic Server Administration Console:

1. Display the Administration Console, as described in Section 3.4.1.
2. Lock the Oracle WebLogic Server configuration, as described in Section 3.4.2.
3. In the left pane, expand **Environment,** then select **Servers.**
4. In the Servers table, click **New.**

   The Create a New Server: Server Properties page is displayed.

5. Enter the following information:
   - 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.

   - For **Listen Address,** to limit the valid addresses for a server instance, enter an IP address or DNS name. Otherwise, URLs to the server can specify any of the host computer's IP address, any DNS name that maps to one of the IP addresses, or the localhost string.
   - 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.

   - Specify whether this server will be a standalone server or will belong to an existing cluster or a new cluster.
     - If this server is to be a standalone server, select **No, this is a stand-alone server.**

   - If this server is to be part of an existing cluster, select **Yes, make this server a member an existing cluster.** Then, select the cluster.

     This option is not shown if there are no existing clusters.

   - If this server is to be part of a new cluster, select **Yes, create a new cluster for this server.**

6. Click **Next.**

   The Review Choices page is displayed.

7. Review the information. If it is correct, click **Finish.**

8. Apply JRF to the Managed Server or cluster as described in Section 18.3.1.

Note that you can also use Fusion Middleware Control to add a Managed Server to a domain. From the Farm menu, choose **Create/Delete Components.** Then, in the Fusion Middleware Components page, select **Create,** then **WebLogic Server.**

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 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.

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 3.5.1.1 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.)

To configure a Managed Server with JRF, use the following command:

```
applyJRF(target='server1', domainDir='/scratch/Oracle/Middleware/user_
projects/domains/domain1')
```

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='cluster', domainDir='/scratch/Oracle/Middleware/user_
projects/domains/domain1')
```

See Also:

- "Java Required Files Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*
- Section I.2.1 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.

Note: For Identity Management components, Oracle Portal, Oracle Forms Services, Oracle Reports, and Oracle Business Intelligence Discoverer, if one or more Managed Servers that you want to include in a cluster is on a remote host, the WebLogic Server home must have the same full path as the WebLogic Server home of the other Managed Servers.

For example, you have a Managed Server on Host A and a Managed Server on Host B, and you want to include them in a cluster. If the WebLogic Server home on Host A is located in /scratch/oracle/Middleware/wlserver_10.3, the WebLogic Server home on Host B must also be located in /scratch/oracle/Middleware/wlserver_10.3.

To create a cluster of two Managed Servers, soa_server1 and soa_server2, take the following steps:

1. From the Farm menu, choose **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 soa_server1 and soa_server2.

6. Click **Create.**

Now, you have a cluster with two members, soa_server1 and soa_server2.

> **See Also:** *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server* for more information about clusters

If you create a server using Fusion Middleware Control, the JRF template is automatically applied.

18.5 Cloning a Middleware Home or Component

You can clone a Middleware home or an Oracle Internet Directory or Oracle Virtual Directory component. **Cloning** is the process of copying an existing entity to a different location while preserving its configuration. Some situations in which cloning Oracle Fusion Middleware is useful are:

- Creating a Middleware home that is a copy of a production, test, or development environment. Cloning enables you to create a new Middleware 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 Middleware homes or components.
- Preparing a "gold" image of a patched home and deploying it to many hosts.

For information about how to clone a Middleware home or component, see Chapter 19.

19 Cloning Oracle Fusion Middleware

You can clone a Middleware home and certain Oracle Fusion Middleware components, such as Oracle Internet Directory and Oracle Virtual Directory.

This chapter includes the following topics:

19.1 Introduction to Cloning

Cloning is the process of copying an existing entity to a different location while preserving its state. Some situations in which cloning Oracle Fusion Middleware is useful are:

- Creating a Middleware home that is a copy of a production, test, or development environment. Cloning enables you to create a new Middleware home with all patches applied to all of the Oracle homes and the WebLogic Server home in a single step. This is in contrast to separately installing and applying any patches to the WebLogic Server home and separate Oracle homes.
- Preparing a "gold" image of a patched Middleware home and deploying it to many hosts.

The cloned entity behaves the same as the source entity. For example, a cloned Middleware home can be deinstalled or patched using the installer. It can also be used as the source for another cloning operation.

19.2 What You Can Clone

You can clone the following, on the same host or a different host:

- Middleware home: You can clone the Middleware home and all of the Oracle homes within the Middleware home. You can clone a Middleware home that contains no Oracle homes.

  You can apply the clone of the Middleware home to the same host or a different host. The clone must be on the same operating system as the source.

- Oracle Internet Directory: You can clone Oracle Internet Directory to the same or a different Oracle instance, to the same or a different Middleware home, on the same host or a different host.
- Oracle Virtual Directory: You can clone Oracle Virtual Directory to the same or a different Oracle instance, to the same or a different Middleware home, on the same host or a different host.

See Section 19.6 for details of considerations and limitations affecting specific components.

19.3 Understanding the Cloning Process

When you clone an entity of Oracle Fusion Middleware, the cloning process takes a snapshot of the information required for cloning. The following topics describe the cloning process:

- Source Preparation Phase
- Cloning Phase

19.3.1 Source Preparation Phase

At the source, you run the createClone command, specifying the Middleware home or component that you want to clone. The program prepares the source for cloning and creates an archive. It also records the file permissions of the Middleware home and the Oracle homes within the Middleware home.

If you are creating a clone of a Middleware home, the archive contains all of the Oracle homes and Oracle WebLogic Server homes in the Middleware home.

If you are creating a clone of Oracle Virtual Directory or Oracle Internet Directory, the archive contains the component configuration.

19.3.2 Cloning Phase

At the destination, you run a clone command, specifying either the Middleware home or component and the destination. The clone program checks to see that the prerequisites are met at the destination. It extracts the files from the archive file and registers the Oracle homes with the installer and registers WebLogic Server homes with the Middleware home.

The clone program then restores the file permissions and relinks any files if that is necessary.

19.4 Cloning Syntax

Cloning uses a Java program, which is located in:

```
(UNIX) ORACLE_HOME/jlib/cloningclient.jar
(Windows) ORACLE_HOME\jlib\cloningclient.jar
```

The syntax is:

```
java [-d64] -jar ORACLE_HOME/jlib/cloningclient.jar cloning_command options
```

Use the -d64 option to the java command if the operating system is SunOS, HP-UX, HPIA64, HPUX-PARisc, or Linux-64 bit.

Note: If you are applying the clone of a Middleware home on a host that does not yet have Oracle Fusion Middleware installed, the host must have JDK 1.6 or higher installed. In addition, ensure that the PATH, CLASSPATH, and JAVA_HOME environment variables point to the JDK.

Copy the cloningclient.jar file from the source host to the target host. You can copy it to any directory.

Cloning Commands

This section describes the cloning commands. The options are described in the tables that follow the syntax.

Note that all cloning commands ask you if you want to continue whenever the `-silent true` option is not used. To continue, you must type `Yes`, which is not case sensitive. Any words other than `Yes` causes the cloning command to return an error. Also note that, in `silent` mode, the command generates an error if you do not provide passwords where they are needed.

Also note that most options have shortcut names, as described in the tables later in this section.

createClone

Creates an archive file of a Middleware home or a component.

The syntax differs depending on what you are cloning.

The syntax for creating a clone of a Middleware home is:

```
createClone -archiveLocation archive_location
            -sourceMWHomeLoc MW_HOME
            [-invPtrLoc Oracle_InventoryLocation]
            [-logDirLoc log_dir_path]
            [-silent {true | false}]
```

See Table 19–1 for a description of the options for creating a clone of a Middleware home.

The following example shows how to create the clone of a Middleware home:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
       -archiveLocation /tmp/mw_clone.jar
       -sourceMWHomeLoc /scratch/Oracle/Middleware1
       -invPtrLoc /scratch/oracle/oraInst.loc
```

The syntax for creating a clone of a component is:

```
createClone -archiveLocation archive_location
            -sourceInstanceHomeLoc src_instance_path
            -sourceComponentName src_component_name
            [-logDirLoc log_dir_path]
            [-silent {true | false}]
```

You cannot create a clone of a component in the same operation as when you create a clone of a Middleware home.

See Table 19–4 for a description of the options for creating a clone of a component.

The following example shows how to create the clone of the Oracle Virtual Directory instance named ovd1 in the Oracle instance located in /scratch/Oracle/Middleware/im_1:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
       -archiveLocation /tmp/ovd1.jar
       -sourceInstanceHomeLoc /scratch/Oracle/Middleware1/im_1
       -sourceComponentName ovd1
```

listCloneArchive

Lists the contents of the archive. The syntax is:

```
listCloneArchive -archiveLocation archive_location [-logDirLoc logdir_path]
```

See Table 19–3 for a description of the options.

The following example shows how to list the contents of the archive:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar listCloneArchive
               -archiveLocation  /tmp/oh_clone.jar
  Middleware home archive # 1 ,   home location =  /scratch/oracle/mwhome_cln0
  Oracle     home archive # 1 ,   home location =  /scratch/oracle/mwhome_cln0/oh_
cln0
```

applyClone

Applies the clone. If you are cloning a Middleware home, you can apply the clone to the same host or a different host. If you are cloning Oracle Virtual Directory or Oracle Internet Directory, you can apply the clone to the same or a different Oracle instance, to the same or a different Middleware home (on the same host or a different host).

The syntax differs depending on what you are cloning.

The syntax for applying the clone of a Middleware home:

```
applyClone -archiveLocation archive_location
            -targetLocation target_location
            [-executeSysPrereqs {true | false}]
            [-invPtrLoc Oracle_InventoryLocation]
            [-logDirLoc log_dir_path]
            [-javaHome java_home_location]
            [-silent {true | false}]
```

See Table 19–2 for a description of the options for applying the clone of a Middleware home.

The following example shows how to apply the clone to the directory /scratch/oracle/MW_Home_clone:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
         -archiveLocation  /tmp/mw_clone.jar
         -targetLocation /scratch/oracle/MW_Home_clone
```

The syntax for applying the clone of a component, such as Oracle Internet Directory or Oracle Virtual Directory is:

```
applyClone -archiveLocation archive_location
          -targetComponentName trgt_component_name
          -targetInstanceHomeLoc trgt_Instance_path
          [-targetInstanceName trgt_Instance_name]
          [-targetOracleHomeLoc trgt_ORACLE_HOME_path]
          [-silent {true | false}]
          [ <Domain Detail> ]
          {<Oracle Internet Directory Detail> | <Oracle Virtual Directory Detail>}

<Domain Detail> =
         -domainHostName domain_host_name
```

```
        -domainPortNo  domain_port_number
        -domainAdminUserName domain_admin_username
        -domainAdminPassword domain_admin_password_file

<Oracle Internet Directory Detail> =
        -namespace oid_namespace
        -dbHostName database_host_name
        -dbPortNo database_port_number
        -dbServiceName databaseServiceName
        -odsSchemaPassword ods_schema_password_file
        -odssmSchemaPassword odssm_Schema_password_file
        -oidAdminPassword oid_admin_password_file
        [-oidNonSSLPort oid_port ]
        [-oidSSLPort oid_ssl_port ]

<Oracle Virtual Directory Detail> =
        [-ovdAdminPort ovd_admin_port ]
        [-ovdHttpPort ovd_http_port ]
        [-ovdLdapPort ovd_ldap_port ]
        [-ovdLdapSPort ovd_ldap_ssl_port ]
```

See Table 19–5 for a description of the options for applying the clone of a Middleware home.

The following example shows how to apply the clone to the Oracle instance im_2 and name the cloned Oracle Virtual Directory instance ovd_cl:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
           -archiveLocation /tmp/ovd1.jar
           -targetOracleHomeLoc /scratch/Oracle/Middleware/Oracle_IM2
           -targetInstanceHomeLoc /scratch/Oracle/Middleware/im_2
           -targetInstanceName im_2
           -targetComponentName ovd_cl
           -domainHostName myhost
           -domainPortNo  7001
           -domainAdminUserName domain_admin_username
           -domainAdminPassword domain_admin_password_file
```

Table 19–1 describes the options for using the createClone command for a Middleware home.

Table 19–1 Options for the createClone Command for a Middleware Home

Options	Shortcut	Description	Mandatory or Optional
-archiveLocation	-al	The absolute path of the archive location. Use this option to specify the location of the archive file to be created with the createClone command. The archive location must not exist, but its parent directory must exist and have write permission.	Mandatory
-invPtrLoc	-invLoc	The absolute path to the Oracle Inventory pointer. Use this option when creating a clone of a Middleware home, if the inventory location is not in the default location, so that the operation can read the Oracle homes present in the inventory. You can use this option only when you use the -sourceMWHomeLoc option. (On Linux, the default location is /etc/oraInst.loc.)	Optional, if the inventory is in the default location. Otherwise, it is mandatory.

Table 19–1 (Cont.) Options for the createClone Command for a Middleware Home

Options	Shortcut	Description	Mandatory or Optional
-logDirLoc	-ldl	The location of an existing directory. A new log file is created in the directory. The default is the system Temp location.	Optional
-silent	NA	Specifies whether the clone operation operates silently. That is, it does not prompt for confirmation. The default is that the operation prompts for confirmation. To specify that it not prompt for confirmation, specify this option with the value of `true`.	Optional
-sourceMWHomeLoc	-smw	The absolute path of the Middleware home. Use this option to specify the Middleware home to be archived. You can only specify one Middleware home.	Mandatory.

Table 19–2 describes the options for using the applyClone command for a Middleware home.

Table 19–2 Options for the applyClone Command for a Middleware Home

Options	Shortcut	Description	Mandatory or Optional
-archiveLocation	-al	The absolute path of the archive location. Use this option to specify the location of the archive file created with the createClone command. The location must exist.	Mandatory
-executeSysPrereqs	-exsysprereqs	Specifies whether the applyClone operation checks the prerequisites of the Oracle homes. 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	-invLoc	The absolute path to the Oracle Inventory pointer. (On Linux, the default location is /etc/oraInst.loc.)	Optional, if the inventory is in the default location. Otherwise, it is mandatory.
-javaHome	-javaHome	If the source Middleware home was installed without JDK and Oracle JRockit and the cloningclient program is launched from the JRE home, specify this option to configure the Middleware home.	Optional
-logDirLoc	-ldl	The location of an existing directory. A new log file is created in the directory. The default is the system Temp location.	Optional
-silent	NA	Specifies whether the clone operation operates silently. That is, it does not prompt for confirmation. The default is that the operation prompts for confirmation. To specify that it not prompt for confirmation, specify this option with the value of `true`.	Optional
-targetLocation	-tl	The absolute path of the target Middleware home. Ensure that the Middleware home directory does not exist at that location. If it does exist, the command returns an error message.	Mandatory

Table 19–3 describes the options for the listCloneArchive command.

Table 19–3 Options for the listCloneArchive Command

Options	Shortcut	Description	Mandatory or Optional
-archiveLocation	-al	The absolute path of the archive location. Use this option to specify the location of the archive file created by the createClone command. The location must exist.	Mandatory
-logDirLoc	-ldl	The location of an existing directory. A new log file is created in the directory. The default is the system Temp location.	Optional

Table 19–4 describes the options for the createClone command for components.

Table 19–4 Options for the createClone Command for Components

Options	Shortcut	Description	Mandatory or Optional
-archiveLocation	-al	The absolute path of the archive location. Use this option to specify the location of the archive file created by the createClone command.	Mandatory
-logDirLoc	-ldl	The location of an existing directory. A new log file is created in the directory. The default is the system Temp location.	Optional
-silent	NA	Specifies whether the clone operation operates silently. That is, it does not prompt for confirmation. The default is that the operation prompts for confirmation. To specify that it not prompt for confirmation, specify this option with the value of `true`.	Optional
-sourceComponentName	-scn	The name of the component to be cloned. For example, if your Oracle Internet Directory component is named oid1, specify oid1.	Mandatory
-sourceInstanceHomeLoc	-sih	The absolute path of the Oracle instance for the source component.	Mandatory

Table 19–5 describes the options for the applyClone command for components.

Table 19–5 Options for the applyClone Command for Components

Options	Shortcut	Description	Mandatory or Optional
-archiveLocation	-al	The absolute path of the archive location. Use this option to specify the location of the archive file created by the createClone command.	Mandatory
-targetComponentName	-tcn	The name of the component to be cloned.	Mandatory
-targetInstanceName	-tin	The name of the target Oracle instance. The name must be unique in the domain.	Optional, if the targetInstanceHome Loc directory exists. In this case, the operation retrieves the name from the configuration.

Table 19–5 (Cont.) Options for the applyClone Command for Components

Options	Shortcut	Description	Mandatory or Optional
-targetInstanceHomeLoc	-tih	The absolute path of the target Oracle instance. If the Oracle instance directory does not exist at that location, the command creates the directory. The same type of component should not already exist in the Oracle instance.	Mandatory
-targetOracleHomeLoc	-toh	The absolute path of the target Oracle home. The target Oracle home must exist and it must be an Identity Management Oracle home. That is, it must contain the binaries for Oracle Internet Directory or Oracle Virtual Directory.	Optional, if the targetInstanceHomeLoc exists. In this case, the operation retrieves the targetOracleHomeLoc from the configuration.
-silent	NA	Specifies whether the clone operation operates silently. That is, it does not prompt for confirmation. The default is that the operation prompts for confirmation. To specify that it not prompt for confirmation, specify this option with the value of `true`.	Optional
Domain-Specific Options			
-domainHostName	-domainhost	The name of the host on which the domain is configured. Use this option if you want to register the component with the domain.	Optional, if you do not want to register the component with the domain.
-domainPortNo	-domainport	The port number of the domain. Use this option if you want to register the component with the domain. The domain port number is listed in the following file as the adminPort. *`ORACLE_INSTANCE`*`/config/OPMN/opmn/instance.properties` For example: adminPort=7001	Optional, if you do not want to register the component with the domain.
-domainAdminUserName	-domainuser	The name of the administrative user for the domain. Use this option if you want to register the component with the domain.	Optional, if you do not want to register the component with the domain.
-domainAdminPassword	-domainpass	The password file for the administrative user for the domain. Use this option if you want to register the component with the domain.	Optional, if you do not want to register the component with the domain.
Oracle Internet Directory Options			
-dbHostName	-dbhost	The host name on which the database is running, which can be found in the tnsnames.ora file.	Mandatory
-dbPortNo	-dbport	The port number of the database listener, which can be found in the tnsnames.ora file.	Mandatory

Table 19–5 (Cont.) Options for the applyClone Command for Components

Options	Shortcut	Description	Mandatory or Optional
-dbServiceName	-dbservice	The service name for the database, which can be found in the tnsnames.ora file.	Mandatory
-namespace	-namespace	The Oracle Internet Directory namespace. For example: dc=us,dc=oracle,dc=com	Mandatory
-odsSchemaPassword	-odspass	The password file for the ODS schema, which is the schema that contains metadata for Oracle Internet Directory. If it is not provided, the operation prompts for a password.	Optional
-odssmSchemaPassword	-odssmpass	The password file for the ODSSM schema, which is used to access server manageability information for Oracle Internet Directory from the database. If it is not provided, the operation prompts for a password.	Optional
-oidAdminPassword	-oidadminpass	The password file for the Oracle Internet Directory administrator. If it is not provided, the operation prompts for a password.	Optional
-oidNonSSLPort	-oidport	The non-SSL port for Oracle Internet Directory. If you do not provide a port number or if the port number you provide is not available, the applyClone operation uses an available port.	Optional
-oidSSLPort	-oidsport	The SSL port for Oracle Internet Directory. If you do not provide a port number or if the port number you provide is not available, the applyClone operation uses an available port.	Optional
Oracle Virtual Directory Options			
-ovdAdminPort	-ovdadminport	The administration port number for Oracle Virtual Directory. If you do not provide a port number or if the port number you provide is not available, the applyClone operation uses an available port.	Optional
-ovdHttpPort	-ovdhttport	The HTTP listener port number for Oracle Virtual Directory.	Optional
-ovdLdapPort	-ovdldapport	The LDAP non-SSL port number for Oracle Virtual Directory. If you do not provide a port number or if the port number you provide is not available, the applyClone operation uses an available port.	Optional
-ovdLdapSPort	-ovdldapsport	The LDAP SSL port number for Oracle Virtual Directory. If you do not provide a port number or if the port number you provide is not available, the applyClone operation uses an available port.	Optional

19.5 Cloning Oracle Fusion Middleware Entities

The general steps for cloning, whether you are cloning Middleware home or a component, are similar. The general steps are described in Section 19.3.

The following sections describe how you clone these entities and view information about a cloning archive:

- Cloning a Middleware Home
- Cloning Oracle Internet Directory
- Cloning Oracle Virtual Directory
- Listing the Contents of an Archive

19.5.1 Cloning a Middleware Home

You can clone a Middleware home, which can contain zero or more Oracle homes and an Oracle WebLogic Server home.

Note:

- The cloning operation archives only those Oracle homes that lie within a Middleware home. It does not clone Oracle homes that are located outside of the Middleware home.
- You can clone only one Middleware home at a time.
- To create an archive of a Middleware home, Oracle WebLogic Server must be installed in the Middleware home.
- On Windows, ensure that no Oracle WebLogic Server processes are running in the source Middleware home.

You can clone a Middleware home and all of its Oracle homes. The createClone operation finds the WebLogic Server home and the Oracle homes contained within the Middleware home. If there are no Oracle homes in the Middleware home, no Oracle homes will be present in the archive.

To clone a Middleware home:

1. At the source Middleware home, execute the createClone command, using the following syntax:

   ```
   java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
            -archiveLocation  archive_location
            -sourceMWHomeLoc MW_HOME
            [-invPtrLoc Oracle_Inventory_location]
   ```

   For example, to clone a Middleware home that is located at /scratch/Oracle /Middleware1, use the following command:

   ```
   java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
          -archiveLocation /tmp/mw_clone.jar
          -sourceMWHomeLoc /scratch/Oracle/Middleware1
          -invPtrLoc /scratch/oracle/oraInst.loc
   ```

2. If you are cloning the Middleware home to a different host, copy the files to that system. Copy the archive, as well as the cloningclient.jar file.

3. At the target, extract the files from the archive using the applyClone command. Use the following syntax:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
         -archiveLocation archive_location
         -targetLocation target_MW_home
```

   For example, to apply the clone to the directory /scratch/oracle/MW_Home_clone, use the following command:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
         -archiveLocation  /tmp/mw_clone.jar
         -targetLocation /scratch/oracle/MW_Home_clone
```

   The Middleware home is restored at /scratch/oracle/MW_Home_clone and all of the Oracle homes are restored under it with the same name as that of source Oracle home name.

19.5.2 Cloning Oracle Internet Directory

You can clone Oracle Internet Directory to the same or a different Oracle instance, to the same or a different Middleware home, on the same host or a different host.

You must apply the clone to an Oracle home that contains the binaries for Identity Management.

To clone Oracle Internet Directory:

1. At the source Middleware home, execute the createClone command, using the following syntax:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
         -archiveLocation  archive_location
         -sourceInstanceHomeLoc ORACLE_INSTANCE
         -sourceComponentName component_name
```

   For example, to clone Oracle Internet Directory instance named oid1 in the Oracle instance located in /scratch/Oracle/Middleware1/im_1, use the following command:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
       -archiveLocation /tmp/oid1.jar
       -sourceInstanceHomeLoc /scratch/Oracle/Middleware1/im_1
       -sourceComponentName oid1
```

2. If you are cloning Oracle Internet Directory to a different host, copy the files to that system. Copy the archive, as well as the cloningclient.jar file.

3. At the target, extract the files from the archive using the applyClone command. Use the following syntax:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
            -archiveLocation archive_location
            [-targetOracleHomeLoc trgt_ORACLE_HOME_path]
            -targetInstanceHomeLoc trgt_ORACLE_INSTANCE_path
            [-targetInstanceName trgt_ORACLE_INSTANCE_name]
            -targetComponentName trgt_component_name
            -namespace oid_namespace
            [-domainHostName domain_host_name]
            [-domainPortNo  domain_port_number]
            [-domainAdminUserName domain_admin_username]
            [-domainAdminPassword domain_admin_password_file]
            -dbHostName database_host_name
            -dbPortNo database_port_number
            -dbServiceName databaseServiceName
            -odsSchemaPassword ods_schema_password_file
            -odssmSchemaPassword odssm_Schema_password_file
            -oidAdminPassword oid_admin_password_file
            [-oidNonSSLPort oid_port]
            [-oidSSLPort oid_ssl_port]
```

For example, to apply the clone to the Oracle instance im_2 and name the cloned Oracle Internet Directory instance oid_cl, use the following command:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
            -archiveLocation /tmp/oid1.jar
            -targetOracleHomeLoc /scratch/Oracle/Middleware/Oracle_IM2
            -targetInstanceHomeLoc /scratch/Oracle/Middleware/im_2
            -targetInstanceName im_1
            -targetComponentName oid_cl
            -namespace dc=us,dc=oracle,dc=com
            -domainHostName myhost
            -domainPortNo  7001
            -domainAdminUserName domain_admin_username
            -domainAdminPassword domain_admin_password_file
            -dbHostName database_host_name
            -dbPortNo 1521
            -dbServiceName orcl
            -odsSchemaPassword ods_schema_password_file
            -odssmSchemaPassword odssm_Schema_password_file
            -oidAdminPassword oid_admin_password_file
```

Note that under certain conditions, you may see the following errors:

```
OID Cloning: Error cleaning replication agreements
OID Cloning: Error deleting replication dn
OID Cloning: Error updating orclreplicaid
```

If you do, take the following steps:

1. Run the following command:

   ```
   ORACLE_HOME/ldap/bin/remtool -pcleanup
   ```

   When prompted, enter the OID host, non-SSL port, and the ODS schema password.

2. Perform an ldapsearch on the root dn for the orclreplicaid value. Use the following command: '

   ```
   ORACLE_HOME/bin/ldapsearch -p port -h host
      -b "" -s base "(objectclass=*)" orclreplicaid
   ```

3. Using the value in obtained in Step 2, perform an ldapdelete, deleting the following dns from Oracle Internet Directory:

   ```
   cn=replication dn, orclreplicaid=<replicaid>, cn=replication configuration
   orclreplicaid=<replicaid>, cn=replication configuration
   ```

   For example:

   ```
   ldapdelete -p port -h host "cn=replication dn,
      orclreplicaid=replicaid, cn=replication configuration"
   ```

4. Set the orclreplicaid value in the root entry to 0. For example:

```
ORACLE_HOME/bin/ldapmodify -p port -h host -f file.ldif
```

The ldif file will have the following contents:

```
dn:
changetype: modify
replace: orclreplicaid
orclreplicaid: 0
```

5. Restart Oracle Internet Directory.

19.5.3 Cloning Oracle Virtual Directory

You can clone Oracle Virtual Directory to the same or a different Oracle instance, to the same or a different Middleware home, on the same host or a different host.

You must apply the clone to an Oracle home that contains the binaries for Identity Management.

To clone Oracle Virtual Directory:

1. At the source Middleware home, execute the createClone command, using the following syntax:

   ```
   java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
            -archiveLocation  archive_location
            -sourceInstanceHomeLoc ORACLE_INSTANCE
            -sourceComponentName component_name
   ```

   For example, to clone Oracle Virtual Directory instance named ovd1 in the Oracle instance located in /scratch/Oracle/Middleware1/im_1, use the following command:

   ```
   java -jar ORACLE_HOME/jlib/cloningclient.jar createClone
          -archiveLocation /tmp/ovd1.jar
          -sourceInstanceHomeLoc /scratch/Oracle/Middleware1/im_1
          -sourceComponentName ovd1
   ```

2. If you are cloning Oracle Virtual Directory to a different host, copy the files to that system. Copy the archive, as well as the cloningclient.jar file.

3. At the target, extract the files from the archive using the applyClone command. Use the following syntax:

   ```
   java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
               -archiveLocation archive_location
               [-targetOracleHomeLoc trgt_ORACLE_HOME_path]
               -targetInstanceHomeLoc trgt_ORACLE_INSTANCE_path
               [-targetInstanceName trgt_ORACLE_INSTANCE_name]
               -targetComponentName trgt_component_name
               [-domainHostName hostname]
               [-domainPortNo  port]
               [-domainAdminUserName domain_admin_username]
               [-domainAdminPassword domain_admin_password_file]
               [-ovdAdminPort ovd_admin_port]
               [-ovdHttpPort ovd_http_port]
               [-ovdLdapPort ovd_ldap_port]
               [-ovdLdapSPort ovd_ldap_ssl_port]
   ```

   For example, to apply the clone to the Oracle instance im_2 and name the cloned Oracle Virtual Directory instance ovd_cl, use the following command:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar applyClone
            -archiveLocation /tmp/ovd1.jar
            -targetOracleHomeLoc /scratch/Oracle/Middleware/Oracle_IM2
            -targetInstanceHomeLoc /scratch/Oracle/Middleware/im_2
            -targetInstanceName im_2
            -targetComponentName ovd_cl
            -domainHostName myhost
            -domainPortNo  7001
            -domainAdminUserName domain_admin_username
            -domainAdminPassword domain_admin_password_file
```

19.5.4 Listing the Contents of an Archive

Use the listCloneArchive command to list the contents of an archive created with the createClone command. Use the following syntax:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar listCloneArchive
         -archiveLocation  archive_location
```

For example:

```
java -jar ORACLE_HOME/jlib/cloningclient.jar listCloneArchive
               -archiveLocation  /tmp/oh_clone.jar
Middleware home archive # 1 ,   home location =  /scratch/oracle/mwhome_cln0
Oracle     home archive # 1 ,   home location =  /scratch/oracle/mwhome_cln0/oh_
cln0
```

19.6 Considerations and Limitations for Cloning

Note the following important additional considerations about cloning:

- Cloning does not carry over all the dependencies of the source Middleware home, WebLogic Server home, and Oracle homes, such as loadable modules or application-specific libraries to the cloned home, because cloning proceeds by copying the Middleware home and the entire source WebLogic Server home and Oracle homes to the destination Middleware 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 cloned home.

  The Oracle home cloned as a part of the Middleware home clones only the binaries. The cloning of Oracle instances outside of Oracle homes is not supported.

- If you created symbolic links to files or applications outside the source WebLogic Server or Oracle home, you must re-create the link manually in the cloned home for your applications to work properly.

- When you clone a Middleware home, only the read-only portions of the Middleware home are cloned. Any user configuration files, such a the user_projects directory, are excluded from the cloned image. The WebLogic Server domain is not cloned.

- If a cloning operation fails, but it results in the Oracle home being registered with Oracle Inventory, you cannot use the same Oracle home in subsequent cloning operations. Either use another directory and name for the Oracle home in subsequent cloning operations or deinstall the Oracle home before attempting another cloning operation.

- If you are applying the clone of a Middleware home on a host that does not yet have Oracle Fusion Middleware installed, the host must have JDK 1.6 or higher installed. In addition, the PATH, CLASSPATH, and JAVA_HOME environment variables must point to the JDK.

- If an Oracle Internet Directory component is cloned with the same database credentials as the source component, the name of the cloned OID component should be different than the source component to avoid conflicts in the OID schema.
- If an Oracle Internet Directory component is cloned with different database credentials from the source component, the name of the cloned Oracle Internet Directory component should be the same as the source component to avoid conflicts in the OID schema.

20

Moving from a Test to a Production Environment

This chapter describes how to move Oracle Fusion Middleware from a test environment to a production environment. You can develop and test applications in a test environment, and then eventually roll out the test applications and, optionally, test data to your production environment. You can also use this approach for testing and rolling out upgrades.

This chapter includes the following topics:

20.1 Overview of Procedures for Moving from a Test to a Production Environment

This chapter describes how to move the following types of installations from a test environment to a production environment:

- Oracle Fusion Middleware Identity Management components including Oracle Internet Directory, Oracle Directory Integration Platform, and Oracle Virtual Directory.
- Oracle SOA Suite
- Oracle WebCenter
- Web Tier components, which are Oracle HTTP Server and Oracle Web Cache
- Oracle Fusion Middleware Classic components, including Oracle Portal, Oracle Forms Services, Oracle Reports, and Oracle Business Intelligence Discoverer.

The general steps are:

1. Install and configure the production environment with the components that you need.

2. Migrate security information, such as users and groups, the identity and policy stores, and credentials.
3. Migrate other data, such as UMS user messaging preferences, data for Oracle WebCenter applications, Oracle Web Cache configuration files. Modify any information that is specific to the new environment such as host name or ports.
4. Deploy applications.

20.2 Moving Identity Management Components to a Production Environment

The following topics describe how to move Identity Management from a test environment to a production environment:

- Moving Identity Management to a New Production Environment
- Moving Identity Management to an Existing Production Environment

In both scenarios, you have performed the following in the test environment:

- Installed a database to be used for Identity Management components such as Oracle Internet Directory, Oracle Directory Integration Platform (which depends on Oracle Internet Directory,) and Oracle Identity Federation.
- Created needed schemas using RCU
- Installed and configured Identity Management, including the following components:
  - Oracle Internet Directory
  - Oracle Virtual Directory
  - Oracle Directory Integration Platform
  - Oracle Identity Federation
  - Oracle Directory Services Manager
  - Oracle Platform Security
  - Oracle Web Services Manager
  - SSL
  - Oracle Access Manager, Release 10.1.4.3, which is used for authentication
- For Oracle Internet Directory, created the desired LDAP trees and entries, in particular, users and groups.
- For Oracle Virtual Directory, created adapters to various data sources, such as LDAP and databases, and you may have configured a Local Store Adapter (LSA) to create local LDAP data, which resides in the local file system. In addition, you may have made other configuration changes such as adding ACLs, changing schemas, the Listener configuration, server configuration, plug-ins, mappings, auditing, logging, and key stores.
- For Oracle Directory Integration Platform, created synchronization profiles to various targets. These profiles are in the form of LDAP entries residing in Oracle Internet Directory.
- For Oracle Identity Federation, configured various trusted identity providers and service providers.

- For Oracle Access Manager, set up authentication with corresponding WebGates configured in the Web tier of the protected applications. The Oracle Access Manager configuration policy and configuration data resides in an LDAP directory.
- For Oracle Platform Security, created an Oracle WebLogic Server domain and used Oracle Platform Security to create security policies and to store credentials in the Credential Store Framework (CSF). Oracle Platform Security is useful for applications such as ADF, WebCenter, and SOA Composite applications.
- For Oracle Web Services Manager, created an Oracle WebLogic Server domain and created Oracle Web Services Manager policies. These policies are also attached to Web services and clients.
- For SSL, configured self-signed certificates. (In a production environment, you use trusted CA-signed certificates.)

20.2.1 Moving Identity Management to a New Production Environment

In this scenario, you have installed Identity Management components, such as Oracle Internet Directory, Oracle Virtual Directory, and Oracle Directory Integration Platform, in a test environment and you want to move them to a production environment that does not exist.

To move this environment to a production environment, perform the following tasks:

- Task 1, "Copy the Database to a New Production Environment"
- Task 2, "Move Oracle Internet Directory to a New Production Environment"
- Task 3, "Move Oracle Virtual Directory to a New Production Environment"
- Task 4, "Move Oracle Directory Integration Platform to a New Production System"
- Task 5, "Move Oracle Identity Federation to a New Production Environment"
- Task 6, "Move Oracle Access Manager to a New Production Environment"
- Task 7, "Move Audit Policies to a New Production Environment"
- Task 8, "Move Oracle Platform Security to a New Production Environment"
- Task 9, "Move Oracle Web Services Manager to a New Production Environment"

Task 1 Copy the Database to a New Production Environment

Some components, such as Oracle Internet Directory, Oracle Directory Integration Platform (which depends on Oracle Internet Directory), and Oracle Identity Federation, require a database.

You can create a duplicate database using the Oracle Database RMAN duplicate command. The duplicate database must be created with a different DBID than the source database, so that it functions entirely independently.

To move an Oracle Database, Release 11*g*, to the production system:

1. On the production system, install the Oracle Database software, but do not create a database. To do this, select **Install database Software only** in the Select Configuration Option screen.
2. On the test environment, edit the tnsnames.ora file, adding an entry for the database on the production environment.

The following shows an example of the tnsnames.ora file. In the example, testDB is the database on the test environment and prodDB is the database on the production environment.

```
testDB =
   (DESCRIPTION =
     (ADDRESS =
       (PROTOCOL = TCP)
       (HOST = 192.168.1.1)
       (PORT = 1521))
         (CONNECT_DATA =
       (SERVER = DEDICATED)
       (SID = testDB)
       )
     )
prodDB=
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = 192.168.2.4)
        (PORT = 1521))
          (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SID = prodDB)
      )
  )
```

3. On the test environment, edit the listener.ora file, adding an entry for the database on the production environment.

   The following shows the added entry:

```
LISTENER_mts =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)
      (HOST = 192.168.2.4)
      (PORT = 1521)(IP = FIRST))
    )
  )
SID_LIST_LISTENER_mts =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = prodDB)
      (ORACLE_HOME = /scratch/oracle/test)
    )
  )
```

4. In the production environment, create a password file in the *ORACLE_HOME*/dbs directory. The sys password must be the same as the database in the test environment. The following command creates the password file:

```
orapwd password=password file=ORACLE_HOME/dbs/orapwproddb
```

5. In the production environment, create a parameter file (pfile) in the *ORACLE_HOME*/dbs directory. The file should contain only the DB_NAME parameter. For example:

```
DB_NAME=prodDB
```

6. In the production environment, set the ORACLE_SID environment variable to point to the production database if it is not already set. Then, start the database in NOMOUNT mode. For example:

   ```
   SQL> STARTUP NOMOUNT PFILE='ORACLE_HOME/dbs/pfile'
   ```

7. On the test system, run the following command:

   ```
   RMAN
   DUPLICATE TARGET DATABASE
     TO prodDB
     FROM ACTIVE DATABASE
     SPFILE
     NOFILENAMECHECK;
   ```

   RMAN automatically copies the server parameter file to the destination host, starts the auxiliary instance with the server parameter file, copies all necessary database files and archived redo logs over the network to the destination host, and recovers the database. Finally, RMAN opens the database with the RESETLOGS option to create the online redo logs.

   > **See Also:** To migrate a Release 10*g* Oracle Database, see the Oracle Database documentation

Task 2 Move Oracle Internet Directory to a New Production Environment

To move Oracle Internet Directory to a new production environment:

1. Clone the Middleware home containing Oracle Internet Directory. See Section 19.5.1. The Oracle homes in the Middleware home are also cloned.
2. Clone Oracle Internet Directory. See Section 19.5.2.
3. If you have configured Oracle Internet Directory replication in the test environment, you must reconfigure it again in the production environment after cloning. Replication configuration is not migrated from the test to the production environment. See "Setting Up Replication" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory.*

Task 3 Move Oracle Virtual Directory to a New Production Environment

To move Oracle Virtual Directory to a new production environment:

1. Clone the Middleware home containing Oracle Virtual Directory. See Section 19.5.1. The Oracle homes in the Middleware home are also cloned.
2. Clone Oracle Virtual Directory. See Section 19.5.3.

Task 4 Move Oracle Directory Integration Platform to a New Production System

To move Oracle Directory Integration Platform to a production system:

1. Install and configure Oracle WebLogic Server, creating a Middleware home, as described in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.
2. Install and configure Oracle Internet Directory, as described in Task 2, "Move Oracle Internet Directory to a New Production Environment".
3. Install and configure Oracle Directory Integration Platform, specifying the information for the production environment, as described in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

Oracle Directory Integration Platform profiles reside in Oracle Internet Directory. If you have correctly moved Oracle Internet Directory to the production system, the profiles are carried over to the production system.

4. If you configured SSL on the test environment, that configuration is not moved to the production environment. You must configure SSL on the production environment.

Task 5 Move Oracle Identity Federation to a New Production Environment

An Oracle Identity Federation setup involves different modules, such as the Credential Store Framework (CSF) for credentials, JPS authorization rules to access CSF, Audit, that are configured, and that you would need to migrate. Because of that, Oracle recommends that you install a new Oracle Identity Federation server, then configure the new instance, as opposed to be moving settings from the test environment to the production environment.

For information, see *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

Task 6 Move Oracle Access Manager to a New Production Environment

To move Oracle Access Manager to a production system:

1. Install and configure Oracle WebLogic Server, creating a Middleware home, as described in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.
2. Migrate the Directory Server from the test environment to the production environment. That is, migrate the o=oblix node. See "Preparing the New Directory Server Instance" in the *Oracle Access Manager Installation Guide.*
3. Remove the entries that are associated with the Identity Server, Policy Manager, and Access Servers. The entries are under the following:

   ```
   obcontainerId=DBAgents,<Configuration DN>
   ```

   Do not delete the container (obcontainerId=DBAgents).
4. Install and configure Oracle Access Manager, specifying the LDAP information for the production environment, as described in the *Oracle Access Manager Administration Guide.*

   Oracle Access Manager stores policy and configuration data in the LDAP directory. If the LDAP directory is correctly configured (for example if you have correctly moved Oracle Internet Directory from the test environment to the production environment), Oracle Access Manager inherits the policy and configuration data from the LDAP directory.
5. On the production system, install the Identity Server and WebPass using new identifiers. For more information, see:

   - "Installing the Identity Server" in the *Oracle Access Manager Installation Guide.*
   - "Installing WebPass" in the *Oracle Access Manager Installation Guide.*

   After installation, take the following steps:

   a. Start the server.

   b. Complete the identity system browser setup. See "Setting Up the Identity System" in the *Oracle Access Manager Installation Guide.*

6. Install the Policy Manager, as described in "Installing the Policy Manager" in the *Oracle Access Manager Installation Guide.* However, do not update the schema because you already updated it when you migrated the Directory Server. Do not configure the authentication scheme because it already exists in the Directory Server.

   Note: After setting up the production Policy Manager, when you log in as the Oracle Access Manager Administrator, you may get the following error:

   ```
   There was a problem obtaining the user ID. One possible reason for
   this is a time difference between the Identity System and Access
   Systems (Policy Manager and Access System Console).
   ```

   To fix this, from the LDAP, delete the cookie encryption key (without changing the CPResponseEncryptionKey) under the o=oblix node, and restart the Identity Server. Note that you should make a backup of the cookie encryption entry into an ldif file before deletion.

7. Complete the browser setup from the Access System Console, adding the Access Server with a new identifier. See "Creating an Access Server Instance in the System Console" in the *Oracle Access Manager Installation Guide* for more information.

   Also see "About the Access Server and Installation" in the *Oracle Access Manager Installation Guide* for additional information.

8. This scenario reuses the existing WebGate identifier for the production WebGates. Take the following steps:

   a. Navigate to the Access System Console and select the Access System Configuration tab.

   b. Select **Host Identifiers.** On the List all host identifiers page, select the host identifier that is used by the test system.

   c. Click **Modify.** Then, add the host name and port for the production Web server to the **Hostname variations** field.

   Note: Resources may become unprotected if you have the same host and port in multiple host identifiers.

   Ensure that only the host identifier used in the policy domain has the host:port in its definition. Remove host:port from other host identifiers.

   d. Click **Save.**

   e. From the Access System Configuration tab, select **Access Gate Configuration**. Then, select the relevant Access Gate.

   f. In the Details for AccessGate page, click **Modify.**

   g. Change the **Hostname** and **Port,** specifying the host name and port of the production Web server.

   h. Change the Preferred HTTP Host, specifying the host name variation that you added in Step c.

i. Associate the WebGate to the newly added production Access Server, as described in "Associating AccessGates and WebGates with Access Servers" in the *Oracle Access Manager Access Administration Guide*.

j. Disable the WebGate temporarily. From the Access System Console, select the Access System Configuration tab, then select **AccessGate Configuration.** Click **Go** to search. From the results, select an AccessGate. Then, click **Modify.** Click **Disabled.** Then, click **Save.**

   You will enable it after you install the Access Server.

9. Install the Access Server using the new identifier that you used while creating the WebGates. See "Installing the Access Server" in the *Oracle Access Manager Installation Guide.*

10. Install the new WebGate. See 'Installing the WebGate" in the *Oracle Access Manager Installation Guide.*

11. Verify entries and delete entries related to the test environment:

    a. From the Identity System Console, select the System Configuration tab, then select **Directory Profiles.** Verify that the respective Directory Profiles are associated with the new Identity Server, Access Server, and Policy Manager.

    b. From the Identity System Console, select the System Configuration tab, then select **Webpass** and delete the entry for the test WebPass.

    c. From the Identity System Console, select the System Configuration tab, then select **Identity Server** and delete the entry for the test Identity Server.

    d. From the Access System Console, select the Access System Configuration tab, then select **Access Server Configuration.** Delete the entry for the test environment Access Server.

12. From the Identity System Console, select the System Configuration tab, then select **Password Policy.** If the host and port are set for **Password Change Redirect URL,** change them to point to the new Identity Server.

13. From the Access System Console, select the Access System Configuration tab, then select **Authentication Management.** Select the authentication scheme for which Challenge redirect is set. Modify **Challenge Redirect** to specify the host and port of the new Web server, if the new authentication WebGate is installed.

14. From the Access System Console, select the Access System Configuration tab, then select **Authentication Management.** Select the authentication scheme for which a password policy is configured. Change the obWebPassURLprefix (if it exists) to accommodate the new host and port of the production Web server on which WebPass is installed, if WebPass and WebGate reside on different Web servers.

    For more information, see "Configuring Password Policies" in the *Oracle Access Manager Identity and Common Administration Guide*.

Task 7 Move Audit Policies to a New Production Environment

To move audit policies to a new production environment, see the following topics in the *Oracle Fusion Middleware Security Guide*:

- "Migrating Audit Policies"
- "Managing Audit Policies"

Task 8 Move Oracle Platform Security to a New Production Environment

To move Oracle Platform Security to a production environment, you migrate the policy store and credential store:

1. Migrate the policy store, using the WLST command `migrateSecurityStore`, as described in "Migrating Policies with the Command migrateSecurityStore" in the *Oracle Fusion Middleware Security Guide*.
2. Migrate the credential store, using the WLST command `migrateSecurityStore`, as described in "Migrating Credentials with the Command migrateSecurityStore" in the *Oracle Fusion Middleware Security Guide*.
3. If you are using Oracle Web Services Manager, migrate audit polices, as described in "Migrating Audit Policies" in the *Oracle Fusion Middleware Security Guide.*

Task 9 Move Oracle Web Services Manager to a New Production Environment

To move Oracle Web Services Manager to a production environment:

1. Move Oracle Platform Security to the production system, as described in Task 8, "Move Oracle Platform Security to a New Production Environment".

   Oracle Web Services Manager depends on Oracle Platform Security and Oracle Fusion Middleware Audit Framework. Oracle Web Services Manager uses Oracle Platform Security for the following:

   - Credential store: Oracle Web Services Manager stores client policy user name and password credentials and keystore passwords in the credential store.
   - Policy store: Oracle Web Services Manager permission-based authorization policies use Oracle Platform Security policy store to look up permissions.
   - Login modules: Oracle Web Services Manager uses Oracle Platform Security login modules for all of its authentication.
   - Keystore configuration. However, the keystores in the test and production environments are typically different.

2. Migrate audit polices, as described in "Migrating Audit Policies" in the *Oracle Fusion Middleware Security Guide.*
3. Migrate policies stored in the MDS Repository:

   a. From the test environment, execute the following WLST commands:

   ```
   exportMetadata(application='wsm-pm',server='server_name',
       docs='/assertiontemplates/**',toLocation='/tmp/owsmexport/')
   exportMetadata(application='wsm-pm',server='server_name',
        docs='/policies/**',toLocation='/tmp/owsmexport/')
   ```

   b. Copy the /tmp/owsmexport directory from the test environment to the production environment.

   c. In the production environment, execute the following WLST commands:

   ```
   deleteMetadata(application='wsm-pm',server='server_name',
        docs='/assertiontemplates/**')
   deleteMetadata(application='wsm-pm',server='server_name',
        docs='/policies/**')
   importMetadata(application='wsm-pm',server='server_name',
        docs='/assertiontemplates/**',fromLocation='/tmp/owsmexport/')
   importMetadata(application='wsm-pm',server='server_name',
        docs='/policies/**',fromLocation='/tmp/owsmexport/')
   ```

   d. If you have custom-built policies, migrate those by copying the jar files from the test to the production environment. The jar files are located in the following directory:

      ```
      DOMAIN_HOME/lib
      ```

   If you have policy attachments, the attachments are migrated with the application if you use the Oracle WebLogic Server cloning feature.

4. Oracle WebLogic Server JAX-WS applications use policies stored in wsm-seed-policies.jar instead of in MDS. Migrate policies stored in wsm-seed-policies.jar, by copying the following file from the test environment to the production environment:

   ```
   ORACLE_HOME/modules/oracle.wsm.policies_11.1.1/wsm-seed-policies.jar
   ```

5. Migrate the keystore from the test environment to the production environment.

   a. Because private keys differ between test and production environments, they do not need to be migrated.

   b. Public keys, intermediate certificates, and root certificates can be migrated. Use the Sun Microsystems java keytool export and import commands to migrate them.

   c. After migration, review the certificates to see if they are applicable in the production environment based on the clients invoking the services.

   d. If the encryption key alias for the production keystore is different than the test environment keystore, then you must update the rcpt-key-alias for all the policies that perform message protection in the policy configuration.

      From Fusion Middleware Control, select the domain. Then, from the WebLogic Domain menu, choose **Web Services,** then **Policies.** Select the policy and click **Edit.** Update the alias.

      See Also: "Managing Horizontal Policy Migration" in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*

20.2.2 Moving Identity Management to an Existing Production Environment

In this scenario, you have installed Identity Management components, such as Oracle Internet Directory, Oracle Directory Integration Platform, and Oracle Web Services Manager, in a test environment and you want to move them to a production environment that already exists.

On the existing production system, you have installed and configured the components. You want to move an application from the test environment to the production environment while retaining its security-related configuration. This requires migrating application-specific data from the test Identity Management environment to the production Identity Management environment.

To move Identity Management to an existing production environment, perform the following tasks:

- Task 1, "Move Oracle Internet Directory to an Existing Production Environment"
- Task 2, "Move Oracle Platform Security to an Existing Production Environment"
- Task 3, "Move Oracle Web Services Manager to an Existing Production Environment"
- Task 4, "Move Oracle Access Manager to an Existing Production Environment"

Task 1 Move Oracle Internet Directory to an Existing Production Environment

To move Oracle Internet Directory to an existing production environment:

1. You may have configured Oracle Platform Security to use the users and groups in the test environment. If you want to migrate the users and groups from the test environment, take the following steps:

   a. Identify the Default Subscriber for the test Oracle Internet Directory instance by running the following command from the test Oracle home:

   ```
   ORACLE_HOME/bin/ldapsearch -h test_oid_host -p test_oid_port
     -D "cn=orcladmin" -w "test_orcladmin_passwd"
     -b "cn=Common,cn=Products,cn=OracleContext"
     -s base "objectclass=*"  orcldefaultsubscriber
   ```

   This query returns a value for the attribute orcldefaultSubscriber. The value is used in following steps as *default subscriber.*

   b. Retrieve the users from test Oracle Internet Directory instance by running the following command from the test Oracle home:

   ```
   ORACLE_HOME/bin/ldapsearch -h test_oid_host -p test_oid_port
     -D "cn=orcladmin" -w "test_orcladmin_passwd"
     -L -b "cn=users, default subscriber"
     -s sub "objectclass=*" * orclguid > users.ldif
   ```

   c. Migrate the users into the production Oracle Internet Directory instance by running the following command from the production Oracle home:

   ```
   ORACLE_HOME/bin/ldapaddmt -h production_oid_host
     -p production_oid_port -D "cn=orcladmin"
     -w "production_orcladmin_passwd" -r -f ldif_filename
   ```

   Specify the -r argument to migrate data and resolve conflicts. The *ldif_filename* is the file you obtained in the previous step.

2. If the test environment is set up as a staging environment to mimic the production environment, Oracle recommends that you set up one-way replication from the production Oracle Internet Directory to the test Oracle Internet Directory to ensure that any users or groups that exist in the production environment are available in the fan-out replica, which can be used to test applications. Fan-out replication also provides the capability to keep the test Oracle Internet Directory synchronized with the production and to replicate any users or groups that are added into production on real-time basis.

   For information about fan-out replication, see the following sections in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*:

   - "Understanding Oracle Internet Directory Replication"
   - "Setting Up a One-Way, Two-Way, or Multimaster LDAP-Based Replication Agreement by Using the Replication Wizard in Fusion Middleware Control"

3. Migrate Resource Access Descriptors (RAD), if you use Oracle Forms Services or Oracle Reports. This procedure assumes that you have migrated the Default Subscriber from the test environment to the production environment, as described in Step 1. It also assumes that the orclGUIDs of the users at the test Oracle Internet Directory are identical to the production Oracle Internet Directory.

   Take the following steps:

   a. Identify the Default Subscriber as described in Step 1a.

b. Retrieve the RADs from the test Oracle Internet Directory instance, by running the following command:

```
ORACLE_HOME/bin/ldapsearch -h test_oid_host -w test_orcladmin_passwd
   -p test_oid_port -D "cn=orcladmin"
   -L -b "cn=Extended Properties,cn=OracleContext, %default subscriber%"
   -s sub "objectclass=*" * orclguid > rads.ldif
```

c. Migrate the RADs into the production Oracle Internet Directory instance, by running the following command:

```
ORACLE_HOME/bin/ldapaddmt -h production_oid_host
   -p production_oid_port -D "cn=orcladmin"
   -w "production_orcladmin_passwd" -r -f ldif_filename
```

Specify the -r argument to migrate data and resolve conflicts. The *ldif_filename* is the file you obtained in the previous step.

Note that this command generates the file add.log in the directory where you run it. Check the add.log file for errors encountered during RADs migration. If there are any errors, fix the errors and rerun the command.

Task 2 Move Oracle Platform Security to an Existing Production Environment

You must move all of the Oracle Platform Security policy and credential store information from the test environment to an existing production environment:

1. Migrate the policy store, using the WLST command `migrateSecurityStore`, as described in "Migrating Policies with the Command migrateSecurityStore" in the *Oracle Fusion Middleware Security Guide*.
2. Migrate the credential store, using the WLST command `migrateSecurityStore`, as described in "Migrating Credentials with the Command migrateSecurityStore" in the *Oracle Fusion Middleware Security Guide*.
3. Users and groups in the production LDAP may differ from that in the LDAP. There is a mapping between Oracle Platform Security application roles and LDAP roles. While the application roles may remain the same, the mapping to LDAP groups can be changed to map to the corresponding LDAP group in the production environment. See "Managing Application Roles" in the *Oracle Fusion Middleware Security Guide.*
4. If you are using Oracle Web Services Manager, migrate audit polices, as described in "Migrating Audit Policies" in the *Oracle Fusion Middleware Security Guide.*

Task 3 Move Oracle Web Services Manager to an Existing Production Environment

To move Oracle Web Services Manager to an existing production environment:

1. Migrate policies for SOA Composite applications, WebCenter, or ADF applications, which are stored in the MDS Repository.

   To do so using Fusion Middleware Control:

   a. On the test environment, select the domain. Then, from the WebLogic Domain menu, choose **Web Services,** then **Policies.**

   b. Select a policy, then click **Export to File.**

   The policy is copied to a file on the test environment.

   c. Click **Save File,** then **OK.**

d. Navigate to the location on your local directory to which you want to save the file and update the filename as desired. Click **Save.**

e. Copy the file to the production environment.

f. On the production environment, select the domain. Then, from the WebLogic Domain menu, choose **Web Services,** then **Policies.**

g. Click **Import from File.** Browse to the file and click **OK.**

h. On test environment, select the domain. Then, from the WebLogic Domain menu, choose **Web Services,** then **Policies.**

i. Click **Web Services Assertion Templates** in the upper right corner of the page.

j. Click **Export to File.**

k. Click **Save File,** then **OK.**

l. Navigate to the location on your local directory to which you want to save the file and update the filename as desired. Click **Save.**

m. On the production environment, select the domain. Then, from the WebLogic Domain menu, choose **Web Services,** then **Policies.**

n. Click **Import from File.** Browse to the file and click **OK.**

o. Click **Web Services Assertion Templates** in the upper right corner of the page.

p. Click **Import from File.** Browse to the file and click **OK.**

To do so using WLST:

a. From the test environment, execute the following WLST commands:

```
exportMetadata(application='wsm-pm',server='server_name',
   docs='/assertiontemplates/assert_template_name',
   toLocation='/tmp/owsmexport/')
exportMetadata(application='wsm-pm',server='server_name',
   docs='/policies/policy_name',toLocation='/tmp/owsmexport/')
```

b. Copy the /tmp/owsmexport directory from the test environment to the production environment.

c. In the production environment, execute the following WLST commands:

```
importMetadata(application='wsm-pm',server='server_name',
  docs='/assertiontemplates/assert_template_name'',
  fromLocation='/tmp/owsmexport/')
importMetadata(application='wsm-pm',server='server_name',
  docs='/policies/policy_name',fromLocation='/tmp/owsmexport/')
```

d. If you have custom-built policies, migrate those by copying the jar files from the test to the production environment. The jar files are located in the following directory:

```
DOMAIN_HOME/lib
```

See Also: "Managing Horizontal Policy Migration" in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*

2. Oracle WebLogic Server JAX-WS applications use policies stored in wsm-seed-policies.jar instead of in MDS. Migrate policies stored in

wsm-seed-policies.jar, by copying the following file from the test environment to the production environment:

```
ORACLE_HOME/modules/oracle.wsm.policies_11.1.1/wsm-seed-policies.jar
```

You can also use the Oracle WebLogic Server Administration Console to migrate these policies.

3. Migrate any policy attachments for a SOA, ADF, or WebCenter application if they have changed since the application was first deployed in the production environment. For example, policy A was initially configured in the test environment with the BASIC 128 algorithm and attached to the HelloWorld application. The application was deployed to the production environment. Then, on the test environment, you changed policy A to use the Basic 129 algorithm.
4. Migrate any policy attachments for a JAX-WS application if they have changed since the application was first deployed.

Task 4 Move Oracle Access Manager to an Existing Production Environment

In this scenario, you have created a policy domain for the application in the test environment. To migrate that to the production environment:

1. In the production environment, use the Oracle Access Manager OAMCfgTool to create the same policy domain for the application. Ensure that the following specify the production environment:

   ```
   web_domain (The Host identifier is derived from this entry)
   protected_uris="uri1,uri2,uri3"
   app_agent_password=password to be provisioned for the WebGate
   ldap_host=hostname of LDAP server
   ldap_port=port of LDAP server
   ldap_userdn=DN of LDAP Admin User
   ldap_userpassword=password of LDAP Admin User
   oam_aaa_host=host of OAM server
   oam_aaa_port=port of OAM server
   ```

   If you are using a uris_file to specify the protected and public URIs in a file, review the file to ensure that you are listed the corrected URIs.
2. If you made other changes to the Oracle Access Manager entities, such as the policy domain, in the test environment, make the same types of changes in the production environment.

20.3 Moving Oracle SOA Suite to a Production Environment

The following topics describe how to move Oracle SOA Suite from a test environment to a production environment:

- Moving Oracle SOA Suite to a New Production Environment
- Moving Oracle SOA Suite to an Existing Production Environment

In both scenarios, you have the performed the following in a test environment:

- Installed Oracle WebLogic Server and created the Middleware home.
- Created the required schemas in the test database using RCU.
- Installed Oracle SOA Suite.
- Configured Oracle SOA Suite using the Configuration Wizard.

- Installed and configured Identity Management components, such as Oracle Internet Directory, Oracle Platform Security, and Oracle Web Services Manager.
- Deployed one or more applications or SOA Composite applications. The applications have internal and external references.
- Changed some configuration settings. For example, you may have changed something in the config directory, in MDS, or another data source.

20.3.1 Moving Oracle SOA Suite to a New Production Environment

In this scenario, you have installed Oracle SOA Suite in a test environment as described in Section 20.3 and you want to move it to a production environment, which does not yet exist.

To move this environment to a production environment, perform the following tasks:

- Task 1, "Install the Software and Perform the Initial Configuration"
- Task 2, "Configure Resource Adapters, Data Sources, and JMS Resources"
- Task 3, "Configure Security in the New Production Environment"
- Task 4, "Deploy Applications That Support Composite References in the New Production Environment"
- Task 5, "Migrate Composite Deployments to the New Production Environment"
- Task 6, "Move Human Workflow to the New Production Environment"
- Task 7, "Move Oracle B2B to the New Production Environment"

Task 1 Install the Software and Perform the Initial Configuration

> **See Also:** See the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite* for information about setting up an enterprise deployment for Oracle SOA Suite

To install the software and perform the initial configuration on the production system:

1. Install Oracle WebLogic Server. See "Install Oracle WebLogic Server and Create the Middleware Home" in *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.
2. Create the required schemas in the production database using RCU. See *Oracle Fusion Middleware Repository Creation Utility User's Guide*.
3. Install Oracle SOA Suite. See "Installation Instructions" in *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.
4. Configure Oracle SOA Suite and create a domain using the Configuration Wizard. See "Configuring Oracle SOA Suite" in *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.
5. Install and configure Identity Management components, such as Oracle Internet Directory, Oracle Platform Security, and Oracle Web Services Manager.

   For information about installing Identity Management components, see *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite.*

   For information about configuring users and groups in Oracle Internet Directory, see "Configuring Users and Groups in the Oracle Internet Directory and Oracle Virtual Directory Authentication Providers" in the *Oracle Fusion Middleware Securing Oracle WebLogic Server.*

6. Create any custom schemas used by your applications. For example, if your application uses a custom schema in the test environment, create the schema in the production environment.

7. Create directory structures for any inbound or outbound files. For example, if you are using a file adapter that reads an inbound file from the /tmp/inbound_msg directory and writes outbound files to the /tmp/outbound_msg directory, create those directories on the production environment. Similarly, if Oracle B2B is using a listening channel that reads inbound messages from the /tmp/inbound directory and writes outbound messages to the /tmp/outbound directory, create those directories.

Task 2 Configure Resource Adapters, Data Sources, and JMS Resources

To configure resource adapters, data sources, and JMS:

1. To configure database adapters, see "Deployment" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

2. To configure AQ adapters, see "Configuring the Data Sources in the Oracle WebLogic Server Administration Console" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

3. To configure JMS adapters, see "Using the Adapter Configuration Wizard to Configure Oracle JMS Adapter" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

   For more information about configuring these resources using the Administration Console, see the following topics in the Oracle WebLogic Server Administration Console Online Help.

   - Configure Resource Adapters
   - Configure JDBC Data Sources
   - Configure JMS System Modules and Add JMS Resources

Task 3 Configure Security in the New Production Environment

You must configure security. The steps you take depends on the configuration of your environment and application. The following steps assume that you are using Oracle Internet Directory, JKS certificates, Oracle Web Services Manager, and Oracle Platform Security:

1. Set up the jpsroot node in Oracle Internet Directory:

   a. Create an LDIF file with the following contents:

   ```
   dn: cn=jpsroot_soa_domain,dc=us,dc=oracle,dc=com
   cn: jpsroot_soa_domain
   objectclass: top
   objectclass: orclcontainer
   ```

   b. Add the node with the directory, using the following command. (*ORACLE_HOME* is the Oracle home for Identity Management.)

   ```
   ORACLE_HOME/bin/ldapadd -h oidhost -p oidport -D cn=orcladmin
     -w "oid Adminpassword" -vf filename.ldif
   ```

2. If necessary, migrate users and groups to the production environment. For example, if you are using the Human Workflow demo, migrate those users:

   a. Export the users and groups from LDAP identity store on the test environment, using the ldapsearch command. This produces an ldif file that you later import into the LDAP identity store in the production 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" "cn=Users,dc=us
```

   b. Import the ldif file that you exported from the test environment into the production 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
```

3. Export any JKS certificates for B2B endpoints from the test environment to the production environment. Then, import them to the production environment. For information about exporting and importing JKS certificates, see Section 7.3.3.

4. Migrate Oracle Web Services Manager policies, which are stored in the MDS Repository:

   a. From the test environment, execute the following WLST commands:

```
exportMetadata(application='wsm-pm',server='server_name',
   docs='/assertiontemplates/**',toLocation='/tmp/owsmexport/')
exportMetadata(application='wsm-pm',server='server_name',
   docs='/policies/**',toLocation='/tmp/owsmexport/')
```

   b. Copy the /tmp/owsmexport directory from the test environment to the production environment.

   c. In the production environment, execute the following WLST commands:

```
deleteMetadata(application='wsm-pm',server='server_name',
   docs='/assertiontemplates/**')
deleteMetadata(application='wsm-pm',server='server_name',
   docs='/policies/**')
importMetadata(application='wsm-pm',server='server_name',
   docs='/assertiontemplates/**',fromLocation='/tmp/owsmexport/')
importMetadata(application='wsm-pm',server='server_name',
   docs='/policies/**',fromLocation='/tmp/owsmexport/')
```

   d. If you have custom-built policies, migrate those by copying the jar files from the test to the production environment. The jar files are located in the following directory:

```
DOMAIN_HOME/lib
```

   If you have policy attachments, the attachments are migrated with the application if you use the Oracle WebLogic Server cloning feature.

5. Import security policies that are related to the Human Workflow application roles from the test environment to the production environment, as described in Task 2, "Move Oracle Platform Security to an Existing Production Environment" in Section 20.2.2.

6. Import the credential store information from the test environment to the production environment, as described in Task 2, "Move Oracle Platform Security to an Existing Production Environment" in Section 20.2.2.

Task 4 Deploy Applications That Support Composite References in the New Production Environment

If your composite application refers to EJBs or Plain Old Java Objects (POJOs), deploy those applications:

1. To deploy EJB Modules, see "Deploy EJB Modules" in the Oracle WebLogic Server Administration Console Online Help.
2. To deploy Enterprise Applications, see "Working with Enterprise Applications" in the Oracle WebLogic Server Administration Console Online Help.

Task 5 Migrate Composite Deployments to the New Production Environment

As you move projects from one environment to another (for example, from testing to production), you typically must modify several environment-specific values, such as JDBC connection strings and host names of various servers. Configuration plans enable you to modify these values using a single text (XML) file called a configuration plan. The configuration plan is created in either Oracle JDeveloper or with WebLogic Scripting Tool (WLST) commands. During process deployment, the configuration plan is used to search the SOA project for values that must be replaced to adapt the project to the next target environment.

For information about migrating composite deployments and using configuration plans, see "Moving SOA Composite Applications to and from Development, Test, and Production Environments" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite.*

Task 6 Move Human Workflow to the New Production Environment

To move Human Workflow:

1. Migrate Human Workflow user metadata, such as views, flex field mappings, or vacation rules, from the test environment to the production environment, using the Human Workflow User Config Data Migrator. The Data Migrator is available as an ant target that can be executed in the command line. It calls a properties file, migration.properties, that you create specifying the input parameters for the migration of data.

   The migration.properties file contains the following input parameters:

   ```
   operationType = {EXPORT | IMPORT}
   objectType = {VIEW | RULE | APPROVAL_GROUP | TASK_PAYLOAD_FLEX_FIELD_MAPPING}
   name = name of VIEW or APPROVAL_GROUP or TASK_PAYLOAD_FLEX_FIELD_MAPPING
   user = username of VIEW or RULE
   group = groupname for RULE
   grantPermission = {true | false}
   migrateAttributeLabel = {true | false}
   override = {true | false}
   skip = {true | false}
   migrateToActiveVersion = {true | false}
   ```

   You use the following script:

   ```
   ORACLE_HOME/bin/ant-t2p-worklist.xml
   ```

   The command has the following format:

   ```
   ant -f ant-t2p-worklist.xml
        -Dbea.home=BEA_HOME
        -Dsoa.home=SOA_HOME
        -Dmigration.properties.file=MIGRATION_PROPERTY_FILE_PATH
        -Dsoa.hostname=SOA_HOSTNAME
   ```

```
-Dsoa.rmi.port=SOA_RMI_PORT
-Dsoa.admin.user=SOA_ADMIN_USER
-Dsoa.admin.password=SOA_ADMIN_PASSWORD
-Drealm=REALM
-Dmigration.file=MIGRATION_FILE
-Dmap.file=MAP_FILE
```

For additional information about the migration utility, see "Using the User Metadata Migration Utility" in the *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management.*

Take the following steps:

a. Ensure that the PATH environment variable contains the JAVA_HOME and ANT_HOME environment variables and that they point to the locations within the Oracle SOA Suite installation.

b. Create a migration.properties file to export user metadata for the worklist application (for example rules, views, flex field mappings, vacation rules) from the test environment. You can create the migration.properties file in any location. Note the following:

- You can migrate only non-protected flex field mappings.
- You can migrate only "protected" attribute labels.
- You can only export one type of data at a time.
- When you are migrating data for a particular user or group, you must migrate them in separate operations.
- You must export attribute labels before you export flex field mappings.

  To export attribute labels, use the following values in the migration properties file:

  ```
  objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
  migrateAttributeLabel = true
  ```

  To export flex field mappings, use the following values in the migration properties file:

  ```
  objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
  migrateAttributeLabel = false
  ```

For example, to export all rules for a given user, the migration.properties file would contain the following:

```
operationType = EXPORT
objectType = RULE
name = ALL
user = username
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

Note that the parameter `group` is left blank when you export rules for a given user.

To export all rules for a given group, the migration.properties file would contain the following:

```
operationType = EXPORT
objectType = RULE
name = ALL
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

Note that the parameter `user` is left blank when you export rules for a given group.

c. Export the data. The following example shows how to invoke the command and specify the parameters:

```
ant -f ant-t2p-worklist.xml
  -Dbea.home=/scratch/oracle/MW_HOME
  -Dsoa.home=/scratch/oracle/MW_HOME/AS11gR1SOA
  -Dmigration.properties.file=migration.properties
  -Dsoa.hostname=hostname -Dsoa.rmi.port=7001
  -Dsoa.admin.user=weblogic
  -Dsoa.admin.password=password
  -Drealm=jazn.com
  -Dmigration.file=/tmp/export_all_userRules.xml
  -Dmap.file=/tmp/export_all_userRules_mapper.xml
```

d. Ensure that the application is deployed to the production system.

e. Create the migration.properties file to import user metadata for the worklist application to the production environment. Note the following

– You can only import one type of data at a time.

– When you are migrating data for a particular user or group, you must migrate them in separate operations.

– You must import attribute labels before you import flex field mappings.

To import attribute labels, use the following values in the migration properties file:

```
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
migrateAttributeLabel = true
```

To import flex field mappings, use the following values in the migration properties file:

```
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
migrateAttributeLabel = false
```

For example, to import all rules for a given user, the migration.properties file would contain the following:

```
operationType = IMPORT
objectType = RULE
name = ALL
user = username
group =
grantPermission = true
migrateAttributeLabel = false
override = true
```

```
skip = true
migrateToActiveVersion = false
```

Note that the parameter `group` is left blank when you import rules for a given user.

To import all rules for a given group, the migration.properties file would contain the following:

```
operationType = IMPORT
objectType = RULE
name = ALL
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

Note that the parameter `user` is left blank when you import rules for a given group.

f. Import the data to the production environment from the file export_all_userRules.xml, which you created in the previous steps. The following example shows how to invoke the command and specify the parameters:

```
ant -f ant-t2p-worklist.xml
  -Dbea.home=/scratch/oracle/MW_HOME
  -Dsoa.home=/scratch/oracle/MW_HOME/AS11gR1SOA
  -Dmigration.properties.file=migration.properties
  -Dsoa.hostname=hostname
  -Dsoa.rmi.port=7001
  -Dsoa.admin.user=weblogic
  -Dsoa.admin.password=password
  -Drealm=jazn.com
  -Dmigration.file=/tmp/export_all_userRules.xml
  -Dmap.file=/tmp/export_all_userRules_mapper.xml
```

Note that if the data, such as rules and views, are attached to the user, then the user should be an available user in the production SOA server.

2. Deploy J2EE Human Task Forms, as you would deploy any .ear file. See Section 9.3.1 for more information.

3. If necessary, update the workflow notification configuration with production mail server and inbound and outbound e-mail accounts. See "Configuring Oracle User Messaging Service" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite.*

Task 7 Move Oracle B2B to the New Production Environment

To move Oracle B2B to a new production environment:

1. Migrate Oracle B2B system configuration parameters by using the Oracle B2B interface to configure the properties. See "Configuring System Parameters" in the *Oracle Fusion Middleware User's Guide for Oracle B2B* for details.

2. Migrate other configuration properties by using the B2B command line, as described in "Setting Properties of b2b-config" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.

3. Move the B2B agreements and trading partners to the production environment:

a. Export the data from the test environment. The following example exports multiple deployed and active agreements:

```
ant -f ant-b2b-util.xml b2bexport -Dtpanames="Acme_GC_Agreement1,
 GC_Acme_Agreement1" -Dactive=true -Dexportfile="/tmp/export.zip"
```

b. Import the data to the production environment. The following example imports the elements in the file /tmp/export.zip:

```
ant -f ant-b2b-util.xml b2bimport -Dlocalfile=true
-Dexportfile="/tmp/export.zip"
```

For more information about these commands, see "B2B Command Line Tools" in the *Oracle Fusion Middleware User's Guide for Oracle B2B.*

4. Configure B2B agreement external endpoints with production locations and credentials, as described in "Configuring Channels" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.
5. If your Oracle B2B environment has been configured with Java callouts, manually migrate the callout library. See "Managing Callouts" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.
6. Deploy the B2B agreements, as described in "Deploying an Agreement" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.

20.3.2 Moving Oracle SOA Suite to an Existing Production Environment

In this scenario, you have a working production environment and want to test changes in your applications or configuration before rolling those changes into the production environment. In the test environment, you have the same environment as described in Section 20.3.

To move Oracle SOA Suite to the existing production system:

1. Migrate any software patches from the test environment to the production environment by cloning the Middleware home, including its Oracle homes, as described in Section 19.5.1.
2. Follow the steps in Task 2 in Section 20.3.1, modifying existing resource adapters, data sources, and JMS resources, if any changes have been made in the test environment and those changes need to be migrated to the production environment.
3. If you have added users and groups in the test environment or modified security policies or credentials, follow the steps in Task 3 in Section 20.3.1 to migrate them to the production environment.
4. If you have modified EJBs or POJOs in the test environment that support the composite references, follow the steps Task 4 in Section 20.3.1 to migrate them to the production environment.
5. If you have modified any information in the configuration plans, copy those changes to the production environment. For more information about configuration plans, see "Moving SOA Composite Applications to and from Development, Test, and Production Environments" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite.*
6. If you have made any changes to Human Workflow in the test environment, follow the steps Task 6 in Section 20.3.1 to migrate them to the production environment.

7. If you have made any changes to Oracle B2B in the test environment, follow the steps in Task 7 in Section 20.3.1 to migrate them to the production environment.

   Note that if you export selective agreements using the tpanames parameter, you must import each zip file individually.

20.4 Moving Oracle Business Activity Monitoring to a Production Environment

The following topics describe how to move Oracle Business Activity Monitoring from a test environment to a production environment:

- Moving Oracle Business Activity Monitoring to a New Production Environment
- Moving Oracle Business Activity Monitoring to an Existing Production Environment

In both scenarios, you have performed the following in the test environment:

- Installed Oracle WebLogic Server and created the Middleware home.
- Created the required schemas in the test database using RCU.
- Installed Oracle SOA Suite and Oracle Business Activity Monitoring.
- Configured Oracle SOA Suite and Oracle Business Activity Monitoring using the Configuration Wizard.
- Optionally configured Oracle WebLogic Server dependent artifacts, such as:
  - BAM Adapter
  - Data sources for the database or JMS
- Configured and populated the identity store for Oracle Business Activity Monitoring users.
- Configured security policies.

> **Note:** The Oracle User Messaging Service (UMS) is used in SOA and BAM scenarios. The functionality and actions in both scenarios are similar, but there are small differences. In particular, for BAM, only the email driver is supported, so the reconfiguration steps for UMS only apply to the email driver. Also, BAM does not make use of the UMS User Preferences in this release. Hence, the userprefs migration in UMS migration does not apply to BAM. See Section 20.5 for details on moving UMS from a test to a production system.

20.4.1 Moving Oracle Business Activity Monitoring to a New Production Environment

In this scenario, you have installed Oracle Business Activity Monitoring in a test environment as described in Section 20.4 and you want to move it to a production environment, which does not yet exist.

Perform the following tasks:

- Task 1, "Export BAM Data and Artifacts from the Test Environment"
- Task 2, "Install the Software and Perform the Initial Configuration"
- Task 3, "Set Up the LDAP Provider"
- Task 4, "Migrate BAM Data and Artifacts to the New Production Environment"

Task 1 Export BAM Data and Artifacts from the Test Environment

On the test environment, take the following steps:

1. Export the ORACLEBAM database schema, using the following commands (*ORACLE_HOME* is the Oracle home for the Oracle Database):

```
ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

create or replace directory directory as 'path';
grant read,write on DIRECTORY directory to oraclebam;
exit;
ORACLE_HOME/bin/expdp userid=oraclebam/bam@connect_id
       directory=directory dumpfile=orabam.dmp
       schemas=oraclebam logfile=oraclebam_date.log
```

   > **See Also:** The "Overview of Oracle Data Pump" and other chapters on Oracle Data Pump in the *Oracle Database Utilities*, which is available at:
   >
   > `http://www.oracle.com/technology/documentation/database.html`

2. Export the BAM users from the LDAP identity store on the test environment, using the ldapsearch command. This produces an ldif file that you later import into the LDAP identity store in the production 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
```

Task 2 Install the Software and Perform the Initial Configuration

To install the software and perform the initial configuration:

1. Install Oracle WebLogic Server. See "Install Oracle WebLogic Server and Create the Middleware Home" in *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.
2. Create the required schemas in the production database using RCU. See *Oracle Fusion Middleware Repository Creation Utility User's Guide*.
3. Install Oracle SOA Suite and Oracle Business Activity Monitoring. See "Installation Instructions" in *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.
4. Configure Oracle SOA Suite and Oracle Business Activity Monitoring and create a domain using the Configuration Wizard. See "Configuring Oracle SOA Suite" in *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.

Task 3 Set Up the LDAP Provider

Set up the LDAP provider for BAM applications on the production environment:

1. Install and configure Oracle Internet Directory.

   For information about installing Oracle Internet Directory, see *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite.*

   For information about configuring users and groups in Oracle Internet Directory, see "Configuring Users and Groups in the Oracle Internet Directory and Oracle Virtual Directory Authentication Providers" in the *Oracle Fusion Middleware Securing Oracle WebLogic Server.*

2. Set up the Oracle Internet Directory Authenticator:

    1. From the Oracle WebLogic Server Administration Console, select **Security Realms**, then **myrealm**, then **Providers.**

        A default Authenticator is configured for the realm.

    2. Click **New** to add a new authenticator.
    3. Enter a name for the provider, such as `OIDAuthenticator` for a provider that authenticates the user to the Oracle Internet Directory.
    4. For **Type,** select **OracleInternetDirectoryAuthenticator.**
    5. Click **OK.**
    6. On the Providers tab, click the newly created OIDAuthenticator.
    7. For **Control Flag**, select **Sufficient** to indicate that if a user can be authenticated successfully by this authenticator, then it should accept that authentication and should not continue to invoke any additional authenticators.
    8. Select the Provider Specific tab.
    9. Enter the details of the LDAP provider.
    10. Click **Save.**
    11. In the Providers tab, reorder the authenticators so that the newly created authenticator is first.

3. Restart the Administration Server and the Managed Server.

Task 4 Migrate BAM Data and Artifacts to the New Production Environment

Migrate BAM data and artifacts to the production environment:

1. Create the BAM JPS root context by importing the ldiff file. The following shows a sample ldiff file:

    ```
    dn: cn=jpsroot_bam_test,dc=us,dc=oracle,dc=com
    cn: jpsroot_bam_test
    objectclass: top
    objectclass: orclcontainer
    ```

2. Migrate the BAM application policy and roles to LDAP using Fusion Middleware Control:

    - From the navigation pane, right-click the domain that contains Oracle Business Activity Monitoring and choose **Security,** then **Security Provider Configuration.**
    - Follow the steps in "Reassociating Domain Stores with Fusion Middleware Control" in *Oracle Fusion Middleware Security Guide.*

3. Import the BAM users from the ldif file that you exported from the test environment into the LDAP provider, such as Oracle Internet Directory, on the production environment. (*ORACLE_HOME* is the Oracle home for Identity Management.)

    ```
    ORACLE_HOME/bin/ldapadd -h production_oid_host -p production_oid_port
     -D "cn=orcladmin" -w "production_orcladmin_passwd" -vf ldif_filename
    ```

4. Import the ORACLEBAM database schema that you exported from the test environment, using the following commands (*ORACLE_HOME* is the Oracle home for the Oracle Database):

   ```
   ORACLE_HOME/bin/impdp userid=system/password dumpfile=ORACLEBAM.DMP
      remap_schema=oraclebam:oraclebam TABLE_EXISTS_ACTION=replace
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"
      alter user oraclebam account unlock;
      alter user oraclebam identified by bam;
   ```

   Note that impdp may report the following errors:

   - `ORA-00959: tablespace <source tablespace> does not exist.`

     You can fix this error by creating the tablespace in the import database before the import or use REMAP_TABLESPACES to change the tablespace referenced in the table definition to a tablespace in the import database.

   - You may see failure with restoring index statistics if you use an Oracle database version earlier than 11.2.0.2. You can work around this issue by rebuilding the index statistics after import.

5. If you are using BAM sensors, which are automatically deployed as part of a composite, deploy the composite, as described in Task 5, "Migrate Composite Deployments to the New Production Environment" in Section 20.3.1.

6. Modify the email server configuration on the production environment, as described in "Configuring Oracle User Messaging Service" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite.*

7. Restart the Oracle Business Activity Monitoring Managed Server.

20.4.2 Moving Oracle Business Activity Monitoring to an Existing Production Environment

To move Oracle Business Activity Monitoring to an existing production environment, perform the following tasks:

- Task 1, "Export BAM Data and Artifacts from the Test Environment"
- Task 2, "Migrate BAM Data and Artifacts to the Production Environment"

Task 1 Export BAM Data and Artifacts from the Test Environment

Export data and artifacts from the test environment:

1. Export BAM artifacts from the test environment using the icommand, which is located in the following directory:

   ```
   (UNIX) ORACLE_HOME\bam\bin\icommand.sh
   (Windows) ORACLE_HOME\bam\bin\icommand.bat
   ```

   For example:

   ```
   icommand -cmd export -type dataobject -all 1 -PERMISSIONS 1 -OWNER 1
     -file dataobject.xml
   icommand -cmd export -type folder -all 1 -PERMISSIONS 1 -OWNER 1
     -file folder.xml
   icommand -cmd export -type report -all 1 -file reports.xml
   icommand -cmd export -type rule -all 1 -file rules.xml
   icommand -cmd export -type ems -all 1 -file ems.xml
   icommand -cmd export -type eds -all 1 -file eds.xml
   ```

In addition to exporting all artifacts of a particular type, you can export individual artifacts. For more information about using the icommand to export artifacts, see "Export" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite.*

2. Export the BAM users from the LDAP identity store on the test environment, using the ldapsearch command. This produces an ldif file that you later import into the LDAP identity store in the production 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
```

Task 2 Migrate BAM Data and Artifacts to the Production Environment

To migrate data and artifacts to the production environment:

1. Deactivate the rules that are set up by default, using Oracle BAM Architect. See "To change the activity status of an alert rule" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.
2. If you have not already done so, set up the LDAP security provider and make it the default provider, as described in Task 3, "Set Up the LDAP Provider" in Section 20.4.2.
3. Import the BAM users from the ldif file that you exported from the test environment into the LDAP provider, such as Oracle Internet Directory, on the production environment. (*ORACLE_HOME* is the Oracle home for Identity Management.)

```
ORACLE_HOME/bin/ldapadd -h production_oid_host -p production_oid_port
 -D "cn=orcladmin" -w production_orcladmin_passwd -vf ldif_filename
```

4. Migrate the BAM application policy and roles to LDAP using Fusion Middleware Control:
   - From the navigation pane, right-click the domain that contains Oracle BAM and choose **Security,** then **Security Provider Configuration.**
   - Follow the steps in "Reassociating Domain Stores with Fusion Middleware Control" in *Oracle Fusion Middleware Security Guide.*
5. Import the Oracle BAM artifacts using the icommand, which is located in the following directory:

```
(UNIX) ORACLE_HOME\bam\bin\icommand.sh
(Windows) ORACLE_HOME\bam\bin\icommand.bat
```

For example:

```
icommand -cmd import -file dataobject.xml -UPDATELAYOUT 1
   -MODE UPDATE -CONTINUEONERROR
icommand -cmd import -file folder.xml -MODE OVERWRITE -PRESERVEOWNER
icommand -cmd import -file reports.xml -MODE OVERWRITE -PRESERVEOWNER
icommand -cmd import -file ems.xml -MODE OVERWRITE
icommand -cmd import -file eds.xml -MODE OVERWRITE
```

6. If you are using BAM sensors, deploy the composite, as described in Task 5, "Migrate Composite Deployments to the New Production Environment" in Section 20.3.1.
7. Start the BAM server.

20.5 Moving Oracle User Messaging Service to a Production Environment

In these scenarios, you have installed Oracle User Messaging Service (UMS) in a test environment and you want to move it to a new production environment. The first example shows how to move the full UMS install from a test to a new production environment; the second example shows how to merge certain UMS (*delta*) changes from a test to an existing production environment.

20.5.1 Moving Oracle User Messaging Service to a New Production Environment

In this scenario, you have set up UMS and all required sub-components, have configured UMS drivers and user preferences in a test environment, and want to establish it in a production environment that does not exist.

Task 1 Migrate UMS-Related Details to the Production Environment

To migrate UMS details to the production environment:

1. Configure the required UMS drivers in the production environment.
   - Use Fusion Middleware Control to configure the User Messaging Service drivers with production driver information.
   - Use the WLST command `deployUserMessagingDriver` to deploy multiple drivers similar to the test environment.

     Note: To see different options for deploying additional drivers, execute `help('deployUserMessagingDriver')` at the `wls:/offline>` prompt.

   - Re-create any custom-created *business terms* in the production environment. This step is essential in order to use the same set of *User Preferences* filter settings in the production environment, and to ensure that filters built with custom business terms are functional.
   - Restart the production environment to apply the changes.
2. Migrate the User Messaging preferences from the test environment to the production environment:

   a. Go to the test machine.

   b. Run the following WLST commands to download the User Messaging preferences from the backend database to the specified `.xml` file:

   ```
   wls:/offline> manageUserMessagingPrefs(operation='download',
       filename='/tmp/userprefs-dump.xml', url='t3://localhost:8001',
       username='weblogic', password='password')
   wls:/offline> exit()
   ```

   Note: In this example, *`weblogic`* is a sample Oracle WebLogic Server administrator user name. Replace it with the real value in your environment. 8001 is the Managed Server port on which UMS is running. Replace it with the appropriate value.

   c. Copy the /tmp/userprefs-dump.xml file to the production machine.

d. Go to production machine.

e. Run the following WLST command to upload the User Messaging preferences from file to the backend database:

```
wls:/offline> manageUserMessagingPrefs(operation='upload',
   filename='/tmp/userprefs-dump.xml', url='t3://localhost:8001',
   username='weblogic', password='password')
```

Note: In the above sample *'weblogic'* is a sample Oracle WebLogic Server administrator user name. Replace it with the real value in your environment. *8001* is the Managed Server port on which UMS is running. Replace it with the appropriate value.

f. Observe the message displayed for successful upload. Exit the WLST command line tool.

Note: To see different options for performing download and upload operations, execute `help('manageUserMessagingPrefs')` at the `wls:/offline>` prompt. Please note that user devices provisioned in the LDAP store are dynamic. The assumption is that both the test and production environments point to the same LDAP store or will be reconfigured to use the same set of information.

Task 2 Validate the Production Environment

Follow these steps to validate the production environment:

1. Test the UMS drivers for send and receive capabilities for supported drivers.
2. Test the successful upload of user messaging preferences by invoking the `http://host:port/sdpmessaging/userprefs-ui` URL. Log in as the desired user and validate that the messaging channels and filters are identical to those in the test environment. Alternatively, send and receive messages that are expected to be delivered based on the User Messaging preferences.

20.5.2 Moving Oracle User Messaging Service to an Existing Production Environment

In this scenario, you have an existing production environment into which you want to move UMS. In this case, you are validating your UMS changes as you merge them into your existing production environment.

Task 1 Migrate UMS-related Details to the Production Environment

Any configuration change must be applied to both test and production environments.

1. Configure the required UMS drivers in the production environment.
   - Use Fusion Middleware Control to configure the User Messaging Service drivers with production driver information.
   - Use the WLST command `deployUserMessagingDriver` to deploy multiple drivers similar to the test environment.

Note: To see different options for deploying additional drivers, execute `help('deployUserMessagingDriver')` at the `wls:/offline>` prompt.

- Re-create any custom-created *business terms* in the production environment. This step is essential in order to use the same set of *User Preferences* filter settings in the production environment, and to ensure that filters built with custom business terms are functional.
- Restart the production environment to apply the changes.

2. Migrate the User Messaging preferences from the test environment to the production environment. Filters cannot be updated or appended to an existing filter set. You must do one of the following:
   - Delete the entire filter set and upload a new set if there are changes made to the filter set in the test environment.
   - Newly created or modified user devices and filters in the test environment must be created or modified using the `http://host:port/sdpmessaging/userprefs-ui` URL in the production environment.

Task 2 Validate the Production Environment

Follow these steps to validate the production environment:

1. Test the UMS drivers for send and receive capabilities for supported drivers.
2. Test the successful upload of user messaging preferences by invoking the `http://host:port/sdpmessaging/userprefs-ui` URL. Log in as the desired user and validate that the messaging channels and filters are identical to those in the test environment. Alternatively, send and receive messages that are expected to be delivered based on the User Messaging preferences.

20.6 Moving Oracle WebCenter to a Production Environment

The following topics describe how to move Oracle WebCenter from a test environment to a production environment:

- Moving Oracle WebCenter to a New Production Environment
- Moving Oracle WebCenter to an Existing Production Environment

In both scenarios, you have the performed the following in a test environment:

- Installed Oracle WebLogic Server.
- Installed Oracle WebCenter.
- Created the required schemas in the test database using RCU.
- Installed and configured Oracle SOA Suite.
- Configured Oracle WebCenter using the Configuration Wizard. You created a domain and Managed Servers and configured Oracle WebCenter Spaces, Oracle WebCenter Portlets, Oracle Discussions, Oracle WebCenter Wiki and Blog Server.
- Installed and configured Oracle Universal Content Management.
- Installed Identity Management components, such as Oracle Internet Directory, Oracle Identity Federation, and Oracle Access Manager.
- Configured the Group Spaces.
- Configured Oracle WebCenter to use LDAP and created some users and groups in the embedded LDAP or an LDAP store.
- Created the required Oracle Platform Security Services policies in the policy store.

- Created the required user credentials in the credential store.
- Created and deployed custom WebCenter applications.
- Created instance data in the WebCenter Spaces application, including creating a group space based on a Community of Interest template, and provisioned services. In addition, created some roles and assigned new members to the roles.

20.6.1 Moving Oracle WebCenter to a New Production Environment

In this scenario, you have installed Oracle WebCenter in a test environment as described in Section 20.6 and you want to move it to a production environment, which does not yet exist. You move WebCenter Spaces applications and custom WebCenter Framework applications.

To move Oracle WebCenter to a new production environment, perform the following tasks:

- Task 1, "Export WebCenter Spaces Applications and Required Data from the Test Environment"
- Task 2, "Export Custom WebCenter Framework Applications from the Test Environment"
- Task 3, "Install and Configure Oracle WebCenter in the Production Environment"
- Task 4, "Import WebCenter Spaces Data and Application to the Production Environment"
- Task 5, "Import Custom WebCenter Framework Applications to the Production Environment"

Task 1 Export WebCenter Spaces Applications and Required Data from the Test Environment

Export WebCenter Spaces applications and data required for the applications from the test environment:

1. Export the required data for WebCenter Spaces applications, including the LDAP identity store, the Content Server, Discussion Forum, Wiki.

   a. Export the users and groups from LDAP identity store, using the ldapsearch command. This produces an ldif file that you later import into the LDAP identity store in the production 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" "cn=Users,dc=us
   ```

   b. Export Oracle Content Server, executing the following commands (*ORACLE_HOME* is the Oracle home for the Oracle Database):

   ```
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

   create or replace directory directory as 'path';
   grant read,write on DIRECTORY directory to user;
   exit;

   ORACLE_HOME/bin/expdp "sys/password@connect_id as sysdba"
      schemas=prefix_OCSERVER directory=directory dumpfile=filename
   ```

> **See Also:** The chapters on Oracle Data Pump in the *Oracle Database Utilities*, which is available at:
>
> `http://www.oracle.com/technology/documentation/database.html`

c. Export the Discussion Forum using the Oracle Database export utility (*ORACLE_HOME* is the Oracle home for the Oracle Database):

```
ORACLE_HOME/bin/export "sys/password@connect_id as sysdba"
   OWNER=prefix_DISCUSSIONS FILE=/tmp/df.dmp statistics=none
```

d. Export the Wiki using the following commands (*ORACLE_HOME* is the Oracle home for the Oracle Database):

```
ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

create or replace directory directory as 'path';
grant read,write on DIRECTORY directory to user;
exit;

ORACLE_HOME/bin/expdp "sys/password@connect_id as sysdba"
   schemas=prefix_WIKI directory=directory dumpfile=filename
```

Refer to "Exporting and Importing WebCenter Spaces Applications for Data Migration" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter* for more information.

2. Migrate Oracle SOA Suite from the test environment to the production environment, as described in Section 20.3.
3. If you want to move all application producer data:

   a. Export the producers using the WSRP and JPDK migration utilities, as described in "Additional Portlet Configuration" in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter.*

   b. Run the migration tool for WebClipping. For example:

```
java -classpath ORACLE_HOME/webcenter/modules/oracle.portlet.server_
11.1.1/wce.jar:
 ORACLE_HOME/webcenter/modules/oracle.portlet.server_11.1.1/wcptg.jar:
 ORACLE_HOME/webcenter/modules/oracle.portlet.server_11.1.1/wcwebdb.jar:
 ORACLE_HOME/oui/jlib/http_client.jar:
 ORACLE_HOME/wlserver_10.3/server/ext/jdbc/oracle/11g/ojdbc5.jar
 oracle.webclipping.util.RepositoryMigrationTool
 -sourceUsername prefix_PORTLET
 -sourcePassword password  -sourceDatabase{ host:port:sid }
 -destUsername prefix_PORTLET  -destPassword password
 -destDatabase { host:port:sid }
```

   Note that you should enter the command on one line.

4. Export the WebCenter Spaces application by using the following WLST commands (*ORACLE_HOME* refers to the Oracle WebCenter Oracle home):

```
connect('username','password','t3://hostname:port')
exportWebCenterApplication(appName,fileName,
exportCustomizations=true, exportSecurity=true, exportData=true)
```

Refer to "Exporting and Importing Custom WebCenter Applications for Data Migration" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter* for details.

Task 2 Export Custom WebCenter Framework Applications from the Test Environment

To export custom WebCenter Framework applications from the test environment:

1. Export the applications' metadata that is stored in the MDS Repository from the source partition to a directory on the file system, using the WLST `exportMetadata` command. (ORACLE_HOME refers to the Oracle WebCenter Oracle home.)

   ```
   exportMetadata(application='app_name', server='server_name',
       toLocation='directory', docs='/**')
   ```

2. Export Portlet customizations and metadata from the test environment, using the following command:

   ```
   exportProducerMetadata(appName='app_name',fileName='filename')
   ```

3. Export the data from the database, using the following commands (*ORACLE_HOME* is the Oracle home for the Oracle Database):

   ```
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

   create or replace directory directory as 'path';
   grant read,write on DIRECTORY directory to user;
   exit;

   ORACLE_HOME/bin/expdp "sys/password@connect_id as sysdba"
      schemas=prefix_WEBCENTER directory=directory dumpfile=filename
   ```

   > **See Also:** The chapters on Oracle Data Pump in the *Oracle Database Utilities*, which is available at:
   >
   > `http://www.oracle.com/technology/documentation/database.html`

Task 3 Install and Configure Oracle WebCenter in the Production Environment

Install and configure Oracle WebCenter in the production environment:

1. Install Oracle WebLogic Server and create the Middleware home. See "Install Oracle WebCenter and Create the Middleware Home" in *Oracle Fusion Middleware Installation Guide for Oracle WebCenter*.
2. Create the required schemas in the production database using RCU. See *Oracle Fusion Middleware Repository Creation Utility User's Guide*.
3. Because Oracle Universal Content Management requires a Web server, install and configure Oracle HTTP Server, as described in the *Oracle Fusion Middleware Installation Guide for Web Tier.*
4. Install Oracle WebCenter, including Oracle Universal Content Management. See "Installing Oracle WebCenter" in *Oracle Fusion Middleware Installation Guide for Oracle WebCenter*.
5. Configure Oracle WebCenter and create a domain using the Configuration Wizard. In the Select Domain Source page, select all components. (JRF is automatically selected.) See "Configuring Oracle WebCenter" in *Oracle Fusion Middleware Installation Guide for Oracle WebCenter*.
6. Associate WebCenter Spaces with an external identity store. See "Configuring the Identity Store" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter.*

Task 4 Import WebCenter Spaces Data and Application to the Production Environment

To import the WebCenter Spaces data and application to the production environment:

1. Import the LDAP identity store, policy, and credential store:

   a. Create authenticators for Identity Management in Oracle WebLogic Server. See "Reassociating the Identity Store with an External LDAP" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter.*

   b. Import the ldif file that you exported from the test environment in Task 1 into the production 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
   ```

   c. Reassociate the policy and credential store using the following command:

   ```
   connect('weblogic','weblogic', 'host:port')
   reassociateSecurityStore(domain="domain_name",
      admin="cn=admin_name",password="password",
      ldapurl="ldap://ldap_host:ldap_port",
      servertype="OID",jpsroot="cn=root_node")
   ```

   Refer to "Exporting and Importing Custom WebCenter Applications for Data Migration" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter* for more information.

2. Import Content Server:

   a. Import the Oracle Content Server data to the production database, using the file you exported in Task 1. Execute the following commands (*ORACLE_HOME* is the Oracle home for the Oracle Database):

   ```
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

   create or replace directory directory as 'path';
   grant read,write on DIRECTORY directory to user;
   exit;

   ORACLE_HOME/bin/impdp "sys/password@connect_id as sysdba"
      remap_schema=testprefix_OCSERVER:prod_prefix_OCSERVER
      DIRECTORY=directory dumpfile=filename
      TABLE_EXISTS_ACTION=REPLACE
   ```

   b. Copy the following directories from the test system to the production system. You can use tar to compress the files from the test system and restore them on the production system:

   ```
   WebCenter_ORACLE_HOME/ucm/vault
   WebCenter_ORACLE_HOME/ucm/weblayout
   ```

3. Import the Discussion Forum:

   a. Connect to the production database using SQLPlus. (*ORACLE_HOME* is the Oracle home for the production database.)

   ```
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"
   ```

   b. Drop the target user:

```
drop user prefix_DISCUSSIONS cascade;
```

c. Create the target user:

```
create user prefix_DISCUSSIONS identified by password
 default tablespace prefix_IAS_DISCUSSIONS
 temporary tablespace prefix_IAS_TEMP;
```

d. Grant connect and resource privileges to the user and exit from SQLPlus:

```
grant connect,resource to prefix_DISCUSSIONS;
exit;
```

e. Import the Discussion Forum data into the production database. You import the file that you exported from the test database in Task 1. (*ORACLE_HOME* is the Oracle home for the production database.)

```
ORACLE_HOME/bin/impdp "sys/password@connect_id as sysdba"
   FROMUSER=testprefix_DISCUSSIONS TOUSER=prod_prefix_DISCUSSIONS
   FILE=filename statistics=none
```

4. Import the Wiki. (*ORACLE_HOME* is the Oracle home for the production database.)

```
ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

create or replace directory directory as 'path';
grant read,write on DIRECTORY directory to user;
exit;

ORACLE_HOME/bin/impdp "sys/password@connect_id as sysdba"
  REMAP_SCHEMA=testprefix_WIKI:prod_prefix_WIKI
  DIRECTORY=directory DUMPFILE=filename TABLE_EXISTS_ACTION=REPLACE
```

5. If you want to move all application producer data:

a. Export the producers using the WSRP and JPDK migration utilities, as described in "Additional Portlet Configuration" in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

b. Run the migration tool for WebClipping. For example:

```
java -classpath ORACLE_HOME/webcenter/modules/oracle.portlet.server_
11.1.1/wce.jar:
 ORACLE_HOME/webcenter/modules/oracle.portlet.server_11.1.1/wcptg.jar:
 ORACLE_HOME/webcenter/modules/oracle.portlet.server_11.1.1/wcwebdb.jar:
 ORACLE_HOME/oui/jlib/http_client.jar:
 ORACLE_HOME/wlserver_10.3/server/ext/jdbc/oracle/11g/ojdbc5.jar
 oracle.webclipping.util.RepositoryMigrationTool
 -sourceUsername test_prefix_PORTLET
 -sourcePassword password  -sourceDatabase{ host:port:sid }
 -destUsername prod_prefix_PORTLET  -destPassword password
 -destDatabase { host:port:sid }
```

Note that you should enter the command on one line.

6. Import the WebCenter Spaces application by using the following WLST commands (*ORACLE_HOME* refers to the Oracle WebCenter Oracle home):

```
connect('username','password','t3://hostname:port')
exportWebCenterApplication(appName,fileName,
exportCustomizations=true, exportSecurity=true, exportData=true)
```

7. Export the WebCenter Spaces application by using the following WLST commands (*ORACLE_HOME* refers to the Oracle WebCenter Oracle home):

   ```
   connect('username','password','t3://hostname:port')
   importWebCenterApplication(appName='appName', fileName='fileName')
   ```

   Refer to "Exporting and Importing Custom WebCenter Applications for Data Migration" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter* for details.

   > **Note:** If you are using the embedded Oracle WebLogic Server LDAP identity store, see "Managing Security" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*.
   >
   > Although secure, the embedded LDAP identity store is not a production-class store and should be replaced with an external LDAP-based identity store such as Oracle Internet Directory for enterprise production environments.

   > **See Also:** "Exporting and Importing WebCenter Applications for Data Migration" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter* for more information.

Task 5 Import Custom WebCenter Framework Applications to the Production Environment

To import custom WebCenter Framework applications to the production environment:

1. Import the applications' metadata from the file you exported in Task 2, to the MDS Repository on the production system, using the WLST `importMetadata` command. (ORACLE_HOME refers to the Oracle WebCenter Oracle home.)

   ```
   importMetadata(application='app_name', server='server_name',
     fromLocation='directory', docs='/**')
   ```

2. Import Portlet customizations and metadata from file you exported in Task 2 to the production environment, using the following WLST command:

   ```
   importProducerMetadata(appName='app_name',fileName='filename')
   ```

3. Import the data from the database, using the following commands (*ORACLE_HOME* is the Oracle home for the Oracle Database):

   ```
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

   create or replace directory directory as 'path';
   grant read,write on DIRECTORY directory to user;
   exit;

   ORACLE_HOME/bin/impdp "sys/password@connect_id as sysdba"
      REMAP_SCHEMA=test_prefix_WEBCENTER:prod_prefix_WEBCENTER
      DIRECTORY=directory dumpfile=filename TABLE_EXISTS_ACTION=REPLACE
   ```

4. If you have not already done so, import the LDAP identity, policy, and credential stores:

   a. Create authenticators for Identity Management in Oracle WebLogic Server. See "Reassociating the identity store with an External LDAP" in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter.*

b. Import the ldif file that you exported from the test environment in Task 1 into the production 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
```

c. Reassociate the policy and credential store using the following WLST commands:

```
connect('weblogic','weblogic', 'host:port')
reassociateSecurityStore(domain="domain_name",
   admin="cn=admin_name",password="password",
   ldapurl="ldap://ldap_host:ldap_port",
   servertype="OID",jpsroot="cn=root_node")
```

5. Migrate the credential and policy store:

a. Copy the following file and name it `jps-config-cred.xml`:

```
DOMAIN_HOME/config/fmwconfig/jps-config.xml
```

b. In the copy of the file, make a copy of the element <serviceInstance provider="ldap.policystore.provider" name="policystore.ldap"> and change the name to `policystore.ldap.s`.

c. Edit the <serviceInstance> element for `policystore.ldap.s`, change the domain name value, the jpsroot value, and ldap url value to the source value. For example:

```
<serviceInstance provider="ldap.policystore.provider"
name="policystore.ldap.s">
  <property value="OID" name="policystore.type"/>
  <property value="bootstrap" name="bootstrap.security.principal.key"/>
  <property value="cn=test_domain" name="oracle.security.jps.farm.name"/>
  <property value="cn=jpsroot_webcenter_test"
name="oracle.security.jps.ldap.root.name"/>
  <property value="ldap://hostname:port" name="ldap.url"/>
</serviceInstance>
```

d. Make a copy of the element <serviceInstance provider="ldap.credentialstore.provider" name="credstore.ldap"> and change the name to `credstore.ldap.s`.

e. Edit the <serviceInstance> element for creditstore.ldap.s, change the domain name, jpsroot, and ldap url value to the source value. For example:

```
serviceInstance provider="ldap.credentialstore.provider"
name="credstore.ldap.s">
  <property value="bootstrap" name="bootstrap.security.principal.key"/>
  <property value="cn=t2ptest_domain"
name="oracle.security.jps.farm.name"/>
  <property value="cn=jpsroot_webcenter_test"
name="oracle.security.jps.ldap.root.name"/>
  <property value="ldap://hostname:port" name="ldap.url"/>
</serviceInstance>
```

f. Make a copy of the element <jpsContext name="default"> and change the name to `source`.

g. Edit the <jpsContext name="default"> element, and remove the reference to idstore.ldap. The following shows the result:

```
<jpsContext name="default">
  <serviceInstanceRef ref="keystore"/>
  <serviceInstanceRef ref="audit"/>
  <serviceInstanceRef ref="policystore.ldap"/>
  <serviceInstanceRef ref="credstore.ldap"/>
</jpsContext>
```

h. Make a copy of the element <jpsContext name="default"> and change the name to `source`.

i. Edit the <jpsContext name="source"> element, change the values for `policystore` and `credstore` to point to the source created in steps b to e.

```
<jpsContext name="source">
  <serviceInstanceRef ref="keystore"/>
  <serviceInstanceRef ref="audit"/>
  <serviceInstanceRef ref="policystore.ldap.s"/>
  <serviceInstanceRef ref="credstore.ldap.s"/>
</jpsContext>
```

j. Find the name of the source folder using the ldapsearch utility:

```
ORACLE_HOME/bin/ldapsearch -h test_host -p port
  -D "cn=admin_name" -w "password" -b ""
  -s sub "cn=application name-*"
```

The folder name shown in the results is in the form: *application name-nnnn*. For example, CustomApp-1646.

k. Find the name of the destination folder using the ldapsearch utility:

```
ORACLE_HOME/bin/ldapsearch -h prod_host -p port
   -D "cn=admin_name" -w "password" -b ""
   -s sub "cn=application name-*"
```

The folder name shown in the results is in the form: *application name-nnnn*. For example, CustomApp-1646.

l. Import the credential store using the following JPS WLST command:

```
migrateSecurityStore(type="folderCred",
   configFile="MW_HOME/user_projects/domains/domain_
name/config/fmwconfig/jps-config-cred.xml",
   src="source",dst="default",overWrite="true",
   srcFolder="source_ folder", dstFolder="destination_folder")
```

m. Find the full name of the application using the ldapsearch utility:

```
ORACLE_HOME/bin/ldapsearch -h test_host -p port
  -D "cn=admin_name" -w "password" -b ""
  -s sub "orclapplicationcommonname=application name*"
```

The application name is returned in the line beginning with `cn=`. For example:

```
cn=CustomApp#V2
```

n. Import the policy store using the following JPS WLST command:

```
migrateSecurityStore(type="appPolicies",
   configFile="MW_HOME/user_projects/domains/domain_
name/config/fmwconfig/jps-config-cred.xml",
src="source" ,dst="default",overWrite="true",
srcApp="full_source_ application name",
dstApp="full_dest_application name")
```

20.6.2 Moving Oracle WebCenter to an Existing Production Environment

In this scenario, you have a working production environment with Oracle WebCenter installed and configured and you want to test changes in your applications or configuration before rolling those changes into the production environment. For example, you have modified a WebCenter Spaces application, you want to deploy a newer version of a WebCenter Framework application, or you have modified existing security policies or configuration.

To move the changes to an existing production environment:

Task 1 Export Group Spaces Data from the Test Environment

To export Group Spaces data from the test environment:

1. Export Group Spaces data from Discussion Forum, using the Oracle Database export utility:

   ```
   ORACLE_HOME/bin/export "sys/password@connect_id as sysdba"
    $discussions_schema/$discussions_password file=discussions_forumid.dmp
    log=jive_forumid.log

   TABLES=jiveforum,jiveThread,jivemessage,jiveForumProp,jiveQuestion,jiveAnswer,j
   iveGateway
    rows=y STATISTICS=None QUERY=\"WHERE forumid \= $forumid\"
   ```

2. Export the Group Spaces data from Wiki, using the owc_wiki_export.sql script. The script is located in the following directory:

   ```
   ORACLE_HOME/wikiserver/owc-wiki/WEB-INF/classes
   ```

   Use the following commands:

   ```
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"

   create or replace directory WC_PUMP_DIR as 'path';
   grant read,write on DIRECTORY WC_PUMP_DIR to user;
   @owc_wiki_export.sql
   ```

3. Export the Group Space from the test environment:

   a. Login to WebCenter Spaces with administrative privileges.

   b. Click the Administration link at the top of the application.

   c. Click the Group Spaces tab.

   d. Click the Group Spaces subtab.

   e. Select the group space required by highlighting the row in the table.

   f. From the Change State drop down, select **Offline.**

   g. Click **Save.**

   h. Click **Export** in the toolbar.

Task 2 Import Group Spaces Data to the Production Environment

To import Group Spaces data to the production environment:

1. Import Group Space data into the Discussion Forum data into the production database. You import the file that you exported from the test database in Task 1. (*ORACLE_HOME* is the Oracle home for the production database.)

   ```
   ORACLE_HOME/bin/impdp "sys/password@connect_id as sysdba"
     file=filename log=df_category.log ignore=y STATISTICS=None
     FROMUSER=test_prefix_DISCUSSIONS TOUSER=prod_prefix_DISCUSSIONS

   imp sys/passwd@dbhost file=T2PTEST_category.dmp log=df_category.log ignore=y
     STATISTICS=None FROMUSER=TEST_DISCUSSIONS TOUSER=PROD_DISCUSSIONS

   imp sys/passwd@dbhost file=T2PTEST_forumid.dmp log=df_forumid.log ignore=y
     STATISTICS=None FROMUSER=TEST_DISCUSSIONS TOUSER=PROD_DISCUSSIONS

   imp sys/passwd@dbhost file=T2PTEST_forumid_perm.dmp log=df_forumid_perm.log
    ignore=y STATISTICS=None FROMUSER=TEST_DISCUSSIONS TOUSER=PROD_DISCUSSIONS
   ```

2. Import Group Space data into Wiki using the owc_wiki_import.sql script. Edit the script, adding the following line before the line `DBMS_DATAPUMP.start_job(dp_handle)`:

   ```
   DBMS_DATAPUMP.METADATA_REMAP(dp_handle,'REMAP_SCHEMA','source','target');
   ```

3. Create the directory WC_PUMP_DIR.

4. Copy the file that you generated when you exported the data from the database in Task 1, to the WC_PUMP_DIR production environment.

5. Run the script:

   ```
   ORACLE_HOME/bin/sqlplus "sys/password as sysdba"
   @owc_wiki_import.sql
   ```

6. Import the Group Space from the test environment:

   a. Login to WebCenter Spaces with administrative privileges.

   b. Click the Administration link at the top of the application.

   c. Click the Group Spaces tab.

   d. Click the Group Spaces subtab.

   e. Select the group space required by highlighting the row in the table.

   f. From the Change State drop down, select **Offline.**

   g. Click **Save.**

   h. Click **Import** in the toolbar and select the exported archive.

7. Import Group Space data from the Content Server, by using WebDAV to drag the folders under the Group Spaces in the test environment to Group Spaces in the production environment.

Task 3 Export WebCenter Group Space Template from the Test Environment

To export the Group Space template from the test environment:

1. Login to WebCenter Spaces with administrative privileges.
2. Click the Administration link at the top of the application.
3. Click the Group Spaces tab.
4. From the Group Spaces tab, select Templates.

5. Click **Export** in the toolbar.

Task 4 Import WebCenter Group Space Template from the Test Environment

To import the Group Space template to the production environment:

1. Login to WebCenter Spaces with administrative privileges.
2. Click the Administration link at the top of the application.
3. Click the Group Spaces tab.
4. From the Group Spaces tab, select Templates.
5. Click **Import** in the toolbar, and select the exported archive.

20.7 Moving the Web Tier to a Production Environment

The following topics describe how to move the Web tier to a production environment:

- Moving the Web Tier to a New Production Environment
- Moving the Web Tier to an Existing Production Environment

20.7.1 Moving the Web Tier to a New Production Environment

The following topics describe how to move the Web tier to a new production environment:

- Moving Oracle HTTP Server to a New Production Environment
- Moving Oracle Web Cache to a New Production Environment

20.7.1.1 Moving Oracle HTTP Server to a New Production Environment

In this scenario, you have installed Oracle HTTP Server in a test environment and you want to move it to a production environment, which does not yet exist. In the test environment, you have:

- Installed Oracle HTTP Server.
- Created an Oracle instance and one or more Oracle HTTP Server component instances.
- Registered the Oracle instance and the Oracle HTTP Server component instances, with an existing JRF-enabled Oracle WebLogic Server Administration Server if you want to manage the components with Fusion Middleware Control.
- Configured mod_wl_ohs to route requests to one or more virtual hosts.
- Configured SSL for one or more virtual hosts.
- Configured Oracle Single Sign-On.
- Configured mod_plsql.
- Configured mod_oradav.
- In addition, you may be using Oracle Access Manager. The Oracle Access Manager Access Servers are not in the test environment. They reside on a separate production system. However, WebGate is running in the test environment.

To move this environment to a production environment, perform the following tasks:

- Task 1, "Create the Oracle Instance and the Components"
- Task 2, "Update Oracle HTTP Server"

- Task 3, "Configure WebGate (Optional)"
- Task 4, "Restart the Processes"

Task 1 Create the Oracle Instance and the Components

In the production environment, install the binary files using cloning and create the Oracle instance and one or more Oracle HTTP Server component instances:

1. Clone the Middleware home containing Oracle HTTP Server. See Section 19.5.1. The Oracle homes in the Middleware home are also cloned.
2. Create the Oracle instance and the Oracle HTTP Server instance, using either the Oracle Universal Installer or the opmnctl command line:
   - To create the instances and components using the Oracle Universal Installer, take the following steps:
     a. Run the following script:

        ```
        (UNIX) ORACLE_HOME/common/bin/config.sh
        (Windows)ORACLE_HOME\common\bin\config.bat
        ```

     b. Follow the instructions in *Oracle Fusion Middleware Installation Guide for Web Tier*.
   - To create the instances and components using the opmnctl command line, take the following steps:
     a. From the command line, go the following directory:

        ```
        (UNIX) ORACLE_HOME/opmn/bin
        (Windows) ORACLE_HOME\opmn\bin
        ```

     b. Create one or more Oracle instances, using the opmnctl `createinstance` command. For example:

        ```
        opmnctl createinstance -oracleInstance /scratch/Oracle/Middleware/inst1
           -adminHost hostname -adminPort 7001
        ```

        This command creates the Oracle instance and, by default, registers the instance with the Oracle WebLogic Server Administration Server.

     c. Create one or more Oracle HTTP Server component instances, using the opmnctl `createcomponent`. For example:

        ```
        opmnctl createcomponent -componentType OHS
            -oracleInstance /scratch/instances/inst1 -componentName ohs1
        ```

Task 2 Update Oracle HTTP Server

Update Oracle HTTP Server:

1. Make a backup copy of the httpd.conf file in the production environment. Then, copy httpd.conf, which is located in the following directory, as well as any non-default include file, from the test environment to a production environment:

   ```
   (UNIX) INSTANCE_HOME/config/OHS/ohs_name
   (Windows) INSTANCE_HOME\config\OHS\ohs_name
   ```

2. In the production environment, 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 non-default directives that were configured at the test environment and have topological (host name, IP address, port number) or other machine-specific information.

3. Copy any custom contents, such as contents that have been changed or added to the htdocs directory, to the production environment Oracle HTTP Server.
4. Optionally, configure SSL:

   a. Make a backup copy of the ssl.conf file in the production environment. Then, copy ssl.conf, which is located in the following directory, from the test environment to the production environment:

   ```
   ORACLE_INSTANCE/config/OHS/component_name
   ```

   b. At the production environment, edit the ssl.conf file, updating the virtual host listen address, if the host name, IP address, or port number is defined, with the new values for the production environment.

   c. If the non-default wallet was used at the test environment:

      a. Export the wallet from the test environment and import it at the production environment. For information about exporting and importing wallets, see Section 7.4.4.

      b. Update the wallet location (if changed) in the ssl.conf file at the production environment.

5. If you are using mod_wl_ohs:

   a. Make a backup copy of the mod_wl_ohs.conf file in the production environment. Then, copy the mod_wl_ohs.conf file, which is located in the following directory, from the test environment to the production environment:

   ```
   ORACLE_INSTANCE/config/OHS/ohs_name
   ```

   b. In the production environment, edit the mod_wl_ohs.conf file, updating the WebLogicHost, WebLogicPort, or WebLogicCluster directives with the host name, IP address, and port number for the production environment. For example:

   ```
   WebLogicHost hostname | [IPaddress]
   WebCluster IPaddress_1:portnum1, IPaddress_2:portnum2, IPaddress_
   3:portnum3, ...
   ```

   c. If SSL is configured for mod_wl_ohs:

      – Export the wallet from the test environment and import it to the production environment. For information about exporting and importing wallets, see Section 7.4.4.

      – Update the TrustedCAFile directive for the wallet location (if it is changed) in the mod_wl_ohs.conf file at the production environment.

   d. At the test environment, inspect the httpd.conf file to determine if some or all of the mod_wl_ohs configuration was defined in that file. If so, update the configuration in the production environment httpd.conf file with the host name, IP address, and port number for the production environment.

6. If you are using mod_osso:

a. Make a backup copy of the mod_osso.conf file in the production environment. Then, copy the mod_osso file, which is located in the following directory, from the test environment to the production environment:

```
ORACLE_INSTANCE/config/OHS/component_name/moduleconf
```

b. Edit mod_osso.conf to add the following information, using values for your deployment. For example:

```
LoadModule osso_module ${ORACLE_HOME}/ohs/modules/mod_osso.so

<IfModule mod_osso.c>

OssoIdleTimeout off
OssoIpCheck on
OssoConfigFile  ORACLE_INSTANCE/config/OHS/ohs_name/osso/osso.conf

#Location is the URI you want to protect
<Location />
require valid-user
#OHS 11g AuthType Osso
#OHS 10g AuthType Basic
AuthType Osso

</Location>

</IfModule>
```

7. If you are using mod_PLSQL:

a. Make a backup copy of the following configuration files in the production environment. Then, copy the files from the test system to the production system.

```
ORACLE_INSTANCE/config/OHS/ohs_name/conf/moduleconf/plsql.conf
ORACLE_INSTANCE/config/OHS/ohs_name/conf/mod_plsql/dads.conf
ORACLE_INSTANCE/config/OHS/ohs_name/conf/mod_plsql/cache.conf
```

b. Edit the files, replacing the values of any properties that contain information about the test environment with the values for the production environment. Usually, you only need to modify the following properties in the dads.conf file:

```
PlsqlDatabaseUserName
PlsqlDatabasePassword
PlsqlDatabaseConnectString
PlsqlNLSLanguage
```

8. If you are using mod_oradav:

a. Make a backup copy of the following configuration file in the production environment. Then, copy the following file from the test system to the production system:

```
ORACLE_INSTANCE/config/OHS/ohs_name/conf/moduleconf/mod_oradav.conf.
```

b. Edit the file, replacing the values of any properties that contain information about the test environment with the values for the production environment. Usually, you only need to modify the following parameters in the mod_oradav.conf file:

```
DAVParam ORACONNECTSN
DAVParam ORAUSER
DAVParam ORACRYPTPASSWORD
```

9. If any changes have been made to auditconfig.xml, which is located in the following directory, make a backup copy of the file in the production environment. Then, copy it from the test environment to the corresponding production environment:

   ```
   ORACLE_INSTANCE/config/OHS/ohs_name/auditconfig.xml
   ```

10. If any changes have been made to *component*-log.xml, make a backup copy of the file in the production environment. Then, copy the file, which is located in the following directory, from the test environment to the production environment:

    ```
    ORACLE_INSTANCE/diagnostics/logs/OHS/component_name
    ```

Task 3 Configure WebGate (Optional)

If you are using Oracle Access Manager, this scenario assumes that the same Oracle Access Manager server is used for both test and production system. In addition, when you clone the Middleware home, the binary files for WebGate are installed in the Oracle home.

1. Run the configureWebGate program located at:

   ```
   WebGate_install_dir/access/oblix/tools/configureWebGate
   ```

2. Provide details for the production environment, using the same configuration settings as the test environment, except for the following:

   - Host name
   - Port
   - Primary HTTP cookie domain (optional)
   - Preferred HTTP Host (depends on environment)

3. Edit the httpd.conf file to update the WebGate information with the values for the production system, modifying the LoadFile, LoadModule, and WebGateInstallDir directives. The following example shows an excerpt of the httpd.conf file, with the items that must be modified in bold.

   ```
   #*** BEGIN WebGate Specific ****

   LoadFile "/webgate_path/access/oblix/lib/libgcc_s.so.1"
   LoadFile "/webgate_path/access/oblix/lib/libstdc++.so.5"

   LoadModule obWebgateModule "/webgate_
   path/access/oblix/apps/webgate/bin/webgate.so"

   WebGateInstalldir "/webgate_path/access"
   ```

Task 4 Restart the Processes

Restart the processes in the Oracle instance:

```
opmnctl stopall
opmnctl startall
```

20.7.1.2 Moving Oracle Web Cache to a New Production Environment

In this scenario, you have installed Oracle Web Cache in a test environment and you want to move it to a production environment, which does not yet exist. In the test environment, you have:

- Installed Oracle Web Cache.
- Configured two or more Oracle instances, each containing an Oracle Web Cache instance.
- Registered the Oracle instances and the Oracle Web Cache instances with an existing JRF-enabled Oracle WebLogic Server Administration Server, if you want to manage the components with Fusion Middleware Control.
- Configured the Oracle Web Cache instances as a Oracle Web Cache cluster.
- Created a site and configured site-to-server mapping.
- Configured Oracle Web Cache to have an SSL-enabled listening address.
- Configured caching rules, and defined filters for request filtering.

To move this environment to a production environment, perform the following tasks:

- Task 1, "Create the Oracle Instances and the Oracle Web Cache Instances"
- Task 2, "Update Oracle Web Cache"

Task 1 Create the Oracle Instances and the Oracle Web Cache Instances

In the production environment, install the binary files by using cloning and create the Oracle instances and Oracle Web Cache instances:

1. Clone the Middleware home containing Oracle Web Cache. See Section 19.5.1. The Oracle homes in the Middleware home are also cloned.
2. Create the Oracle instances and components. Create the Oracle instances on different hosts, each containing an Oracle Web Cache instance.

   To create the instances and components using the Oracle Universal Installer, take the following steps:

   a. Run the following script:

   ```
   (UNIX) ORACLE_HOME/common/bin/config.sh
   (Windows)ORACLE_HOME\common\bin\config.bat
   ```

   b. Follow the instructions in *Oracle Fusion Middleware Installation Guide for Web Tier*.

   To create the instances and components using the command line, take the following steps:

   a. From the command line, go the following directory:

   ```
   (UNIX) ORACLE_HOME/opmn/bin
   (Windows) ORACLE_HOME\opmn\bin
   ```

   b. Create the Oracle instances, using the opmnctl `createinstance` command. For example:

   ```
   opmnctl createinstance -oracleInstance /scratch/Oracle/Middleware/inst1
      -adminHost hostname -adminPort 7001
   ```

   This command creates the Oracle instance and, by default, registers the instance with the Oracle WebLogic Server Administration Server.

c. Create the Oracle Web Cache instances, using the opmnctl `createcomponent` command. For example:

```
opmnctl createcomponent -componentType WebCache
    -oracleInstance /scratch/Oracle/Middleware/inst1
    -componentName webcache1
```

3. Register the Oracle instances, along with all of its components, with the Administration Server, using the opmnctl `registerinstance` command. For example:

```
opmnctl registerinstance -adminHost admin_server_host
     -adminPort admin_server_port -adminUsername username
     -adminPassword password
     -oracleInstance ORACLE_INSTANCE_dir -oracleHome ORACLE_HOME_dir
     -instanceName Instance_name -wlserverHome Middleware_Home
```

Task 2 Update Oracle Web Cache

For each Oracle Web Cache instance, take the following steps:

1. Copy the webcache.xml file, which is located in the following directory, from the test environment to a temporary location:

```
(UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
(Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
```

2. Make the following changes to webcache.xml in the temporary location:

   - If the Web Cache Administration password at the production environment is different from the password at the test environment:

     – Copy the value of PASSWORDHASH attribute of the <USER TYPE="INVALIDATION"> element from the webcache.xml file for the production environment Web Cache instance and replace the current value of corresponding PASSWORDHASH attribute in this temporary webcache.xml.

     – Copy the value of PASSWORDHASH attributes of the <USER TYPE="MONITORING"> element from the webcache.xml file for the production environment Web Cache instance and replace the current value of corresponding PASSWORDHASH attribute in this temporary webcache.xml.

   - Update the NAME and PORT attributes of each <HOST> and <VIRTUALHOSTMAP> elements with the new host name or IP address and port number of the origin servers at the production environment.

   - For each <CACHE> element in webcache.xml, change the following, substituting the values that correspond to the host where the production environment Oracle Web Cache instance is located:

     – Update the NAME, ORACLE_HOME and HOSTNAME attributes.

     – Search for and replace the Oracle instance path.

       Note: Update this information one Oracle Web Cache instance at a time. Do not do a global search and replace, because other Oracle Web Cache instances might be configured in a different Oracle instance running at a different path.

- For each <LISTEN> element, update IPADDR (if it is configured other than `ANY`) and PORT (if Oracle Web Cache uses different ports at the production environment).
- Update the wallet location (if different) for a SSL-enabled listen address. The wallet location is specified within the <WALLET> element for each SSL listen port.
- Update the USERID and GROUPID attributes of the <IDENTITY> element.
- In the <OSWALLET> element, update the wallet location (if different on the production environment) for the original servers. This is the wallet used by Oracle Web Cache to talk to an SSL-enabled origin server).

3. Copy the edited webcache.xml to the following location on the production system:

```
(UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name
(Windows) ORACLE_INSTANCE\config\WebCache\webcache_name
```

4. If any changes have been made to auditconfig.xml, copy it from the test environment to the corresponding production environment.

```
ORACLE_INSTANCE/config/WebCache/webcache_name/auditconfig.xml
```

5. If any changes have been made to *component*-log.xml, first, edit the file to update the log path, and then copy it from the test environment to the corresponding production environment.
6. If any changes have been made to the Oracle Web Cache error pages, which are located in the following directory, copy the error pages from the test environment to the production environment location:

```
(UNIX) ORACLE_INSTANCE/config/WebCache/webcache_name/files
(Windows) ORACLE_INSTANCE\config\WebCache\webcache_name\files
```

7. If a non-default wallet was used at the test environment for either an SSL-enabled listen address or an OSwallet, or both, export the wallets from the test environment and import them at the production environment. For information about exporting and importing wallets, see Section 7.4.4.

20.7.2 Moving the Web Tier to an Existing Production Environment

In this scenario, you have a working production environment and want to test changes in your applications or configuration before rolling those changes into the production environment.

- For Oracle HTTP Server, follow the steps in Section 20.7.1.1.
- For Oracle Web Cache, follow the steps in Section 20.7.1.2.

20.8 Moving Classic Components to a Production Environment

In this scenario, you have installed Oracle Fusion Middleware Classic components, including Oracle Portal, Oracle Forms Services, Oracle Reports, and Oracle Business Intelligence Discoverer, in a test environment and you want to move it to a production environment.

The following topics describe how to move these components from a test environment to a production environment:

- Moving Classic Components to a New Production Environment

- Moving Classic Components to an Existing Production Environment

In both scenarios, you have performed the following in a test environment:

- Installed a database to be used for these components.
- Created schemas needed by the components, using RCU.
- For Oracle BI Discoverer, installed an additional database to be used for the End User Layer (EUL), Discoverer catalog, and OLAP catalog.
- Installed Oracle WebLogic Server and created a Middleware home.
- Installed and configured Identity Management, including Oracle Internet Directory and Oracle Single Sign-On, and a database for Identity Management data.
- Installed and configured Oracle Portal, Oracle Forms Services, Oracle Reports, and Oracle BI Discoverer.
- For Oracle Portal:
  - Created users and groups and assigned page access permissions to the groups.
  - Created new page groups.
  - Created new templates.
  - Created new pages and added contents, such as items and portlets, to the pages.
  - Customized pages, layouts, items, and portlets.
  - Registered producers (database, Web, and WSRP) and customized the portlet from the producers.
  - Registered external applications.
- Set up Forms applications.
- Configured Oracle Reports instances and created connections to the database.
- For Oracle BI Discoverer:
  - For Discoverer Plus, created a new workbook with parameters, calculations, conditions, and totals. Saved the workbook.
  - For Discoverer Viewer, opened the workbook created in Discoverer Plus and performed some formatting, sorting, exporting, and drilling.
  - For Discoverer Plus OLAP, created a new workbook in Discoverer Plus OLAP with custom members, custom expressions, and saved selections. Saved the workbook.
  - For Viewer OLAP, opened the workbook created in Discoverer Plus OLAP and performed some operations such as exporting, linking and unlinking layouts.

20.8.1 Moving Classic Components to a New Production Environment

In this scenario, you have installed Oracle Portal, Oracle Forms Services, Oracle Reports, and Oracle Business Intelligence Discoverer in a test environment and you want to move the components to a production environment that does not exist.

Although this section describes how to move all of the components to a production environment, you can choose to move only some of them.

To move this environment to a production environment, perform the following tasks:

- Task 1, "Copy the Database to the New Production Environment"
- Task 2, "Install and Configure the Components"
- Task 3, "Move Oracle Portal to the New Production Environment"
- Task 4, "Move Oracle Forms Services to the New Production Environment"
- Task 5, "Move Oracle Reports to the New Production System"
- Task 6, "Move Oracle Business Intelligence Discoverer to the New Production Environment"

Task 1 Copy the Database to the New Production Environment

Move the database to the production system by using the Oracle Database RMAN utility.

For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*, which is available at:

`http://www.oracle.com/technology/documentation/database.html`

See Appendix D for the schemas used by each component.

Task 2 Install and Configure the Components

To install and configure the components:

1. Install and configure Oracle WebLogic Server, creating a Middleware home, as described in *Oracle Fusion Middleware Installation Guide for Oracle Portal, Forms, Reports and Discoverer*.

   Alternatively, you can clone the Middleware home, as described in Section 19.5.1.

2. Install and configure the components, as described in the *Oracle Fusion Middleware Installation Guide for Oracle Portal, Forms, Reports and Discoverer*. For Oracle Portal, this includes installing Oracle Internet Directory and Oracle Single Sign-On Release 10.1.3.4.

   For Oracle Portal, specify the credentials to connect to Oracle Internet Directory at the Configure Components screen.

Task 3 Move Oracle Portal to the New Production Environment

To move Oracle Portal configuration to a new production environment:

1. Create a transport set on the test instance that contains the list of page groups to be migrated. For information about creating a transport set, see "Creating Transport Sets" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal.*

2. Export the data from the test environment, as described in "Exporting Data" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal.*

3. On the production environment, create a database link to the test environment, as described in "Creating a Database Link" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal.*

4. Before moving data from a source portal you must first register the portal. Once registered, the source portal can be selected and used to specify the data source in the Transport Sets. See "Register a Source Portal" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal.*

5. Before importing your objects, the contents of the transport set must first be moved to the transport set tables on the target system. You do this by acquiring the transport set from the test environment, using the registered database link described in Step 1. For information about acquiring the transport set, see "Moving Data to the Target System" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal.*

6. Import the data, as described in "Import in Oracle Portal" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal.*

7. Migrate users and groups from the LDAP directory in the test environment to the LDAP directory in the production environment, as described in "Migrating Users and Groups" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Portal.*

8. Import the external applications list using the SSOMig utility:

   a. Run ssomig in export mode on the test system. The command creates a dump file: For example:

   ```
   ssomig -export -s orasso -p orasso_schema_password
    -c tns_alias_for_sso_schema
    -log_d directory_where_dump_needs_to_be_created
    -log_f ssomig.log -d ssomig.dmp
   ```

   b. Run ssomig in import mode on the production system, specifying the dump file created in the previous step. For example:

   ```
   ssomig -import -overwrite -s orasso -p orasso_schema_password
    -c tns_alias_for_sso_schema -d ssomig.dmp
    -log_d directory_where_dump_is_located -discoforce
   ```

9. For the following files, copy any customizations that you want to maintain from the test environment file to the production environment file:

   ```
   DOMAIN_HOME/config/fmwconfig/servers/WLS_
   PORTAL/applications/portal/configuration/portal_plsql.conf
   DOMAIN_HOME/config/fmwconfig/servers/WLS_
   PORTAL/applications/portal/configuration/portal_dads.conf
   DOMAIN_HOME/config/fmwconfig/servers/WLS_
   PORTAL/applications/portal/configuration/appConfig.xml
   ```

10. If you modified any configuration files, restart the Managed Server WLS_ PORTAL.

Note that when Oracle WebCenter or Oracle Portal is moved from test to production using export and import, portlet customizations are included in transport set. You do not need to take any additional steps.

Task 4 Move Oracle Forms Services to the New Production Environment

To move Oracle Forms Services to the production environment:

1. Stop the processes running in the Oracle instance and the Managed Servers in the production environment, using the following commands:

   ```
   ORACLE_INSTANCE/bin/opmnctl stopall
   MW_HOME/user_projects/domains/domain_name/bin/stopManagedWebLogic.sh
               managed_server_name admin_url username password
   ```

2. Copy the Oracle Forms Services application files (.fmx, . mmx, and .plx) from the test environment to the production environment. The location of the files may be specified in the Forms environment configuration file, default.env.

Note that if the files are in a shared network location, you do not need to copy them to the production environment. Instead, add the location to the default.env file.

3. Migrate the application-related data from the test environment to a database in the production environment using database migration tools.
4. Create entries in the SQLnet configuration files to refer to the database in the production environment.
5. Forms applications have single sign-on user names and passwords mapped to the database connect strings. This information is stored in Oracle Internet Directory. Migrate the Forms RAD data from Oracle Internet Directory in the test environment to Oracle Internet Directory in the production environment. See Step 3 in Task 2, "Move Oracle Internet Directory to a New Production Environment".
6. Copy any customizations in the following files that you want to maintain from the test environment file to the production environment file:

Type of File	Location
Forms application configuration	`DOMAIN_HOME/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.1/config/formsweb.cfg`
Forms server configuration	`DOMAIN_HOME/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.1/config/default.env`
Forms HTML template	`ORACLE_INSTANCE/config/FormsComponent/forms/server/base.htm` `ORACLE_INSTANCE/config/FormsComponent/forms/server/basejpi.htm`
WebUtil configuration	`ORACLE_INSTANCE/config/FormsComponent/forms/server/webutil.cfg`
WebUtil HTML template	`ORACLE_INSTANCE/config/FormsComponent/forms/server/webutiljpi.htm` `ORACLE_INSTANCE/config/FormsComponent/forms/server/webutilbase.htm`
Forms OHS directives configuration	`ORACLE_INSTANCE/config/OHS/OHS_name/moduleconf/forms.conf`

If you modified the Oracle HTTP Server forms.conf file, restart Oracle HTTP Server:

```
opmnctl restartproc ias-component=ohs_name
```

7. Copy the following files from the test environment to the production environment:

Type of File	Location
Forms application configuration client-side downloadable pluggable contents	These files are user customizations such as images and are in a location accessible to a Web browser.
Forms trace configuration	`ORACLE_INSTANCE/config/FormsComponent/forms/server/ftrace.cfg`
Forms applications ear	`ORACLE_HOME/forms/j2ee/formsapp.ear`
JVM Controllers configuration	`ORACLE_INSTANCE/config/FRComponent/frcommon/tools/jvm/jvmcontrollers.cfg`

Type of File	Location
FMA configuration	*ORACLE_INSTANCE*/config/FormsComponent/forms/search_replace.properties *ORACLE_INSTANCE*/config/FormsComponent/forms/converter.properties
Forms utilities-specific configuration wrapper shell scripts	UNIX: *ORACLE_INSTANCE*/bin/frmbld.sh *ORACLE_INSTANCE*/bin/frmcmp.sh *ORACLE_INSTANCE*/bin/frmplsqlconv.sh *ORACLE_INSTANCE*/bin/frmxmlsg.sh *ORACLE_INSTANCE*/bin/frmcmp_batch.sh *ORACLE_INSTANCE*/bin/frmf2xml.sh *ORACLE_INSTANCE*/bin/frmxml2f.sh *ORACLE_INSTANCE*/bin/frmxmlv.sh Windows: *ORACLE_HOME*\bin\frmbld.bat *ORACLE_HOME*\bin\frmcmp.bat *ORACLE_INSTANCE*\bin\frmplsqlconv.bat *ORACLE_INSTANCE*\bin\frmxmlsg.bat *ORACLE_INSTANCE*\bin\frmcmp_batch.bat *ORACLE_INSTANCE*\bin\frmf2xml.bat *ORACLE_INSTANCE*\bin\frmxml2f.bat *ORACLE_INSTANCE*\bin\frmxmlv.bat

For the Forms utilities-specific configuration wrapper shell scripts, replace any occurrences of the Oracle home and Oracle instance with the details for the production environment.

8. Start the components in the instance and start the Managed Server, using the following commands:

```
ORACLE_INSTANCE/bin/opmnctl startall
MW_HOME/user_projects/domains/domain_name/bin/startManagedWebLogic.sh
    managed_server_name admin_url
```

9. If you made customizations to the Forms J2EE application ear file, such as overriding the default Forms servlet access URL, custom deploy the Forms J2EE application ear file and create servlet aliases similar to test environment in the Forms J2EE application web.xml file.

Task 5 Move Oracle Reports to the New Production System

To move Oracle Reports to the production environment:

1. For the following Oracle Reports Server configuration files, merge changes made from the test environment to the production environment files. Note that you cannot just copy the files from the test environment to the production environment, because they may have environment-specific information such as Oracle home and Oracle instance names or locations and port numbers.

Type of File	Location
Reports standalone server configuration	*ORACLE_INSTANCE*/config/ReportsServerComponent/*server_name*/rwserver.conf *ORACLE_INSTANCE*/config/ReportsServer/*server_name*/jdbcpds.conf *ORACLE_INSTANCE*/config/ReportsServer/*server_name*/xmlpds.conf *ORACLE_INSTANCE*/config/ReportsServer/*server_name*/textpds.conf *ORACLE_INSTANCE*/config/ReportsServer/*server_name*/rwnetwork.conf *ORACLE_INSTANCE*/config/ReportsServer/*server_name*/component-logs.xml *ORACLE_INSTANCE*/config/ReportsServer/*server_name*/logging.xml

Type of File	Location
Reports in-process server and servlet configuration	DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/cgicmd.dat DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/rwservlet.properties DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/rwserver.conf DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/jdbcpds.conf DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/xmlpds.conf DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/textpds.conf DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/rwnetwork.conf DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/logging.xml DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/logmetadata.xml DOMAIN_HOME/servers/server_namestage/reports/reports/configuration/jazn-data.xml
Reports Tools configuration	ORACLE_INSTANCE/config/ReportsTools/rwbuilder.conf ORACLE_INSTANCE/config/ReportsTools/rwnetwork.conf ORACLE_INSTANCE/config/ReportsTools/jdbcpds.conf ORACLE_INSTANCE/config/ReportsTools/xmlpds.conf ORACLE_INSTANCE/config/ReportsTools/textpds.conf ORACLE_INSTANCE/config/ReportsTools/component-logs.xml ORACLE_INSTANCE/config/ReportsTools/logging.xml
Reports Bridge configuration	ORACLE_INSTANCE/config/ReportsBridge/bridge_name/rwbridge.conf ORACLE_INSTANCE/config/ReportsBridge/bridge_name/rwnetwork.conf ORACLE_INSTANCE/config/ReportsBridge/bridge_name/component-logs.xml ORACLE_INSTANCE/config/ReportsBridge/bridge_name/loggin.xml
Reports shell scripts	(UNIX) ORACLE_INSTANCE/config/reports/bin/rw*.sh (Windows) ORACLE_INSTANCE\config\reports\bin\rw*.bat (UNIX) ORACLE_INSTANCE/config/reports/bin/reports.sh (Windows) ORACLE_INSTANCE\config\reports\bin\reports.bat (UNIX) ORACLE_INSTANCE/config/reports/bin/namingservice.sh (Windows) ORACLE_INSTANCE\config\reports\bin\namingservice.bat

2. For the following Oracle Fusion Middleware configuration files, which are related to Oracle Reports Server configuration files, merge changes made from the test environment to the production environment files. Note that you cannot just copy the files from the test environment to the production environment, because they may have environment-specific information such as Oracle home and Oracle instance names or locations and port numbers.

Type of File	Location

Forms and Reports common files	`Font setup, aliasing, subsetting, embedding:`
	ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin/Uifont.ali
	`Printer configuration (UNIX only):`
	ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin/uiprint.txt
	`Toolkit configuration, encoding (UNIX only):`
	ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin/uTk2Motif.rgb
	`PPD files (UNIX only):`
	ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon//tk/admin/PPD/*
	`AFM files (UNIX only):`
	ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin/AFM/*

EM Agent configuration	*ORACLE_INSTANCE*/EMAGENT/*emagent_name*sysman/emd/target.xml
OPMN configuration	*ORACLE_INSTANCE*/config/OPMN/opmn/opmn.xml
JPS configuration	*DOMAIN_HOME*/config/fmwconfig/jps-config.xml *DOMAIN_HOME*/config/fmwconfig/jps-config-jse.xml *DOMAIN_HOME*/config/fmwconfig/system-jazn-data.xml

3. If you created additional Oracle Reports Server component instances in the test environment, create these in the production environment using opmnctl.

4. For resources related to Oracle Reports Server, take the following actions:

   - Copy any fonts used in the test environment from the directory specified by environment variable REPORTS_FONT_DIRECTORY to production environment. By default, they are in *ORACLE_INSTANCE*/reports/fonts.

   - Migrate the Common UNIX Printing System (CUPS) printing configuration to the production environment, if applicable. Refer to the following for more information:

     http://www.cups.org/

   For more information about using CUPS with Oracle Reports, see "Enhanced Printing on Linux Using CUPS" in the *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services.*

5. For Reports definition files and data tables, take the following actions:

   - Copy the reports files, such as RDF, JSP, REP, and XML files, used in the test environment to the production environment.

   - Deploy JSP Web reports to the production environment in the following location:

     ```
     DOMAIN_HOME/servers/WLS_REPORTS/stage/reports/reports/web.war
     ```

   - Migrate Reports-specific data tables that are referred to in the RDF files to the database in the production environment using database migration tools, such as the Oracle Database export and import utilities.

6. For Reports job-related configuration files, take the following actions:

   - Copy Reports Server cache files to the following location in the production environment:

     ```
     ORACLE_INSTANCE/reports/cache
     ```

- For Reports scheduled job information, copy the server data (*server_name*.dat) file to the following location in production environment:

  ```
  ORACLE_INSTANCE/reports/server
  ```

  Note that because the server name is generated automatically when it is created and the .dat file is named with the server name, the name of the .dat file differs between the test environment and the production environment. Depending on whether it is a standalone server or an in-process server, the name takes one of the following forms:

  ```
  ReportsServer_hostname_instanceName
  rep_wls_reports_hostname_instanceName
  ```

  Change the name of the file to reflect the host name and Oracle instance name in the production environment.

7. If the job repository or job status repository is configured in the database, you must create the same schemas in the production environment database and migrate the data:

   a. Use the following script:

   ```
   ORACLE_HOME/reports/admin/sql/rw_job_repos.sql
   ```

   b. Migrate any data from the test database for the schemas RW_JOBS, RW_SERVER_JOB_QUEUE, and RW_SERVER_QUEUE to production database using database migration tools, such as the Oracle Database export and import utilities.

8. Migrate any user and reports server security policy information. See "Securing Oracle Reports" in the *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*.

9. If you use Oracle Internet Directory as the identity and policy store, migrate the Forms RAD data from Oracle Internet Directory in the test environment to Oracle Internet Directory in the production environment. See Step 3 in Task 2, "Move Oracle Internet Directory to a New Production Environment".

10. If you used JAZN-XML-based identity and policy store in the test environment, migrate them to the LDAP in the production environment. You can use the WLST command `migrateSecurityStore`, as described in "Migrating Policies with the Command migrateSecurityStore" in the *Oracle Fusion Middleware Security Guide*.

11. Migrate the credential store, using the WLST command `migrateSecurityStore`, as described in "Migrating Credentials with the Command migrateSecurityStore" in the *Oracle Fusion Middleware Security Guide*.

12. Migrate any database proxy users to the production database using database cloning tools.

13. If any Reports plug-ins are registered, copy the corresponding .jar files to the production environment and add the path to the files to the environment variable REPORTS_CLASSPATH.

Task 6 Move Oracle Business Intelligence Discoverer to the New Production Environment

To move Oracle BI Discoverer to the new production environment:

1. If you have changed the user preferences, copy the following files from the test environment to the production environment:

```
ORACLE_INSTANCE/config/PreferenceServer/disco-comp-name/.reg_key.dc
ORACLE_INSTANCE/config/PreferenceServer/disco-comp-name/pref.txt
ORACLE_INSTANCE/config/PreferenceServer/disco-comp-name/defaults.txt
```

2. If you have changed the Oracle BI Discoverer settings, copy following files from the test environment to the production environment:

   ```
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/configuration.xml
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/configuration-previe
   w.xml
   ```

   In configuration.xml, change the values of the following elements to reflect the production environment:

   - applicationURL
   - oracleInstance
   - discovererComponentName

3. If you have changed the server configuration files, copy the following file from the test environment to the production environment:

   ```
   ORACLE_INSTANCE/config/OPMN/opmn/opmn.xml
   ```

4. Copy the following file from the test environment to the production environment:

   ```
   ORACLE_INSTANCE/config/OHS/ohs_name/moduleconf/module_disco.conf
   ```

   Change the values of the following elements to reflect the production environment:

   - WebLogicCluster. Valid only if a cluster exists.
   - WebLogicHost
   - WebLogicPort

5. Copy the following files from the test environment to the production environment:

   ```
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/base-descktop.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/blstyles.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/dc-blaf-review.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/dc-blaf.xsd
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/dc-blaf.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/minimal-desktop.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/minimal-pda.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/oracle-desktop.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/oracle-pda.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/pocketPC.xss
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/simple-desktop.xss
   ```

```
DOMAIN_HOME/servers/WLS_
DISCO/stage/discoverer/11.1.1.1.0/discoverer/configuration/swan-desktop.xss
```

6. Copy some or all of the files in the following directory, depending on which files you use:

   ```
   DOMAIN_HOME/servers/WLS_
   DISCO/stage/discoverer/11.1.1.1.0/discoverer/discoverer.war/custom_logos/
   ```

   The files that are used are listed in the configuration.xml file.

7. If you want to use the same database service entries, copy the following file from the test environment to the production environment:

   ```
   ORACLE_HOME/network/admin/tnsnames.ora
   ```

8. Migrate the DISCOVERER schema from the test environment to the production environment. You can use the Oracle Database export and import utilities to migrate the schema.

   Note that if you choose to use the same database for test and production, you do not need to migrate the data.

9. Migrate the EUL data from the test environment to the production environment:

   a. Create the EUL user and an empty EUL on the production database. See "How to create an End User Layer in a new database user" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Discoverer.*

   b. Migrate the EUL schema from the test database by using the Discoverer Administrator to export the schema from the test database and import it into the database in the production environment. For more information, see "About Using the Discoverer Export Wizard and Import Wizard" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Discoverer*.

   c. Run the eul5_id.sql script to give the new EUL a unique reference number. Then, grant the entire Discoverer end user community access to the EUL. The script is located in:

      ```
      ORACLE_ HOME/discoverer/util/eul5_id.sql
      ```

      For more information, see "Creating and Maintaining End User Layers" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Discoverer*.

10. Migrate the catalog data from the test environment to the production environment:

    a. Install the catalog in the production OLAP database, using the following command:

       ```
       java -classpath d4o.jar oracle.dss.d4o.administration.D4OCommand install
         -h hostname -po port -sid sid -su "sys as sysdba"
         -sp password -p d4osys-password -t users
       ```

    b. Authorized users in the production OLAP database, using the following command:

       ```
       java -classpath d4o.jar oracle.dss.d4o.administration.D4OCommand
        authorize -h hostname -po port -sid sid -p d4osys-password -u user
       ```

   c. Export the Discoverer catalog from the test database and import it into the database in the production environment by using the OLAP command utility. For more information see "Using the Discoverer Plus OLAP Command Line Utility to Manage the Discoverer Catalog" in the *Oracle Fusion Middleware Configuration Guide for Oracle Business Intelligence Discoverer*.

11. Migrate Portlet data from the test Discoverer metadata repository to the production Discoverer metadata repository:

   a. Use the Oracle Database export and import utilities.

      Note that you may need to perform the import multiple times to ensure that parent tables are populated before child tables. Use the following order to avoid SQL errors: PTM5_PARTITION, PTM5_PORTLET, PTM5_VERSION, PTM5_INSTANCE, PTM5_SCHEDULE, PTM5_CACHE,PTM5_CUSTOMINFO.

   b. Modify the Portlet Provider URL in the Portal to point to the new production setup.

12. Migrate PStore data:

   a. Delete the default encryption key from the table WWSSO_PS_CONFIGURATION_INFO_T.

   b. Migrate the PStore data for the Discoverer metadata repository using Oracle Database export and import utilities.

      Note that the user names and schema names must be the same in the production environment as in the test environment.

20.8.2 Moving Classic Components to an Existing Production Environment

In this scenario, you have installed Oracle Portal, Oracle Forms Services, Oracle Reports, and Oracle Business Intelligence Discoverer in a test environment and you want to move the components to a production environment that already exists.

To move to an existing production environment, perform the following tasks:

- Task 1, "Move Oracle Portal to an Existing Production Environment"
- Task 2, "Move Oracle Forms Services to an Existing Production Environment"
- Task 3, "Move Oracle Reports to an Existing Production Environment"
- Task 4, "Move Oracle Business Intelligence Discoverer to an Existing Production Environment"

Task 1 Move Oracle Portal to an Existing Production Environment

This scenario assumes that you have made changes to Oracle Portal in the test environment, such as adding pages, adding content to pages, creating new users and groups, and assigning page access permissions for newly created pages to new users and groups.

To move Oracle Portal to an existing production environment, take the steps described in Task 3, "Move Oracle Portal to the New Production Environment" in Section 20.8.1.

Task 2 Move Oracle Forms Services to an Existing Production Environment

To move Oracle Forms Services to the existing production environment:

1. Copy the Oracle Forms Services application files (.fmx, . mmx, and .plx) from the test environment to the production environment. The location of the files may be specified in the Forms environment configuration file, default.env.

   Note that if the files are in a shared network location, you do not need to copy them to the production environment. Instead, add the location to the default.env file.

2. Make any necessary configuration changes as described in "Deploying Your Application" in the *Oracle Fusion Middleware Forms Services Deployment Guide.*

3. Restart the components:

   ```
   opmnctl stopall
   opmnctl startall
   ```

Task 3 Move Oracle Reports to an Existing Production Environment

To move Oracle Reports to an existing production environment, take the same steps as described in Task 5, "Move Oracle Reports to the New Production System" in Section 20.8.1.

Task 4 Move Oracle Business Intelligence Discoverer to an Existing Production Environment

In this scenario, you primarily use the test environment to create EULs for developing a business area without compromising the performance of production systems.

To move Oracle BI Discoverer to an existing production environment:

1. Migrate the configuration files that are listed in Steps 1 and 5 in Task 6, "Move Oracle Business Intelligence Discoverer to the New Production Environment".

2. Migrate the DISCOVERER schema from the test environment to the production environment. You can use the Oracle Database export and import utilities to migrate the schema.

   Note that if you choose to use the same database for test and production, you do not need to migrate the data.

3. Migrate the EUL schema from the test environment to the production environment by using the Oracle database export and import utilities to export the schema from the test database and import it into the database in the production environment.

   Note that the user names and schema names must be the same in the production environment as in the test environment.

Part IX

Appendixes

This part contains the following appendixes:

A

Oracle Fusion Middleware Command-Line Tools

This appendix summarizes the command-line tools that are available in Oracle Fusion Middleware.

Table A–1 Oracle Fusion Middleware Command-Line Tools

Command	Path	Description
bulkdelete	UNIX: *ORACLE_HOME*/ldap/bin/bulkdelete.sh Windows: *ORACLE_HOME*\ldap\bin\bulkdelete.bat	Delete a subtree efficiently in Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
bulkload	UNIX: *ORACLE_HOME*/ldap/bin/bulkload.sh Windows: *ORACLE_HOME*\ldap\bin\bulkload.bat	Create Oracle Internet Directory entries from data residing in or created by other applications. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
bulkmodify	UNIX: *ORACLE_HOME*/ bin/bulkmodify Windows: *ORACLE_HOME*\bin\bulkmodify	Modify a large number of existing Oracle Internet Directory entries in an efficient way. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
catalog	UNIX: *ORACLE_HOME*/ldap/bin/catalog.sh Windows: *ORACLE_HOME*\ldap\bin\catalog.bat	Add and delete catalog entries in Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
chgiphost	UNIX: *ORACLE_HOME*/chgip/scripts/chpiphost.sh Windows: *ORACLE_HOME*\chgip\scripts\chpiphost.bat	Changes the network configuration of Oracle HTTP Server and Oracle Web Cache. **See:** Section 14.1.2
eulbuilder.jar	UNIX: *ORACLE_HOME*/bin/eulbuilder.jar Windows: *ORACLE_HOME*\bin\eulbuilder.jar	Discoverer EUL Java command-line interface. Create and manipulate Discoverer EULs without installing Oracle Discoverer Administrator. **See:** *Oracle Business Intelligence Discoverer EUL Command Line for Java User's Guide*
iasua	UNIX: *ORACLE_HOME*/upgrade/iasua.sh Windows: *ORACLE_HOME*\upgrade\iasua.bat	Oracle Fusion Middleware Upgrade Assistant. **See:** *Oracle Fusion Middleware Upgrade Planning Guide*
frmcmp	UNIX: *ORACLE_HOME*/bin/frmcmp.sh Windows: *ORACLE_HOME*\bin\frmcmp.exe	Start Form Compiler to generate a form. **See:** Oracle Forms Services Online Help
ldapadd	UNIX: *ORACLE_HOME*/bin/ldapadd Windows: *ORACLE_HOME*\bin\ldapadd	Add entries, their object classes, attributes, and values to Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*

Table A–1 (Cont.) Oracle Fusion Middleware Command-Line Tools

Command	Path	Description
ldapaddmt	UNIX: *ORACLE_HOME*/bin/ldapaddmt Windows: *ORACLE_HOME*\bin\ldapaddmt	Add entries, their object classes, attributes, and values to Oracle Internet Directory. Like `ldapadd`, except supports multiple threads for adding entries concurrently. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldapbind	UNIX: *ORACLE_HOME*/bin/ldapbind Windows: *ORACLE_HOME*\bin\ldapbind	Determine if you can authenticate a client to a server. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldapcompare	UNIX: *ORACLE_HOME*/bin/ldapcompare Windows: *ORACLE_HOME*\bin\ldapcompare	Match attribute values you specify in the command line with the attribute values in the Oracle Internet Directory entry. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldapdelete	UNIX: *ORACLE_HOME*/bin/ldapdelete Windows: *ORACLE_HOME*\bin\ldapdelete	Remove entire entries from Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldapmoddn	UNIX: *ORACLE_HOME*/bin/ldapmoddn Windows: *ORACLE_HOME*\bin\ldapmoddn	Modify the DN or RDN of an Oracle Internet Directory entry. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldapmodify	UNIX: *ORACLE_HOME*/bin/ldapmodify Windows: *ORACLE_HOME*\ bin\ldapmodify	Perform actions on attributes in Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldapmodifymt	UNIX: *ORACLE_HOME*/bin/ldapmodifymt Windows: *ORACLE_HOME*\bin\ldapmodifymt	Modify several Oracle Internet Directory entries concurrently. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldapsearch	UNIX: *ORACLE_HOME*/bin/ldapsearch Windows: *ORACLE_HOME*\bin\ldapsearch	Search and retrieve specific entries in Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldifmigrator	UNIX: *ORACLE_HOME*/ bin/ldifmigrator Windows: *ORACLE_HOME*\bin\ldifmigrator.bat	Migrate data from application-specific repositories into Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
ldifwrite	UNIX: *ORACLE_HOME*/ bin/ldifwrite Windows: *ORACLE_HOME*\bin\ldifwrite.bat	Convert to LDIF all or part of the information residing in an Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidcmprec	UNIX: *ORACLE_HOME*/bin/oidcmprec Windows: *ORACLE_HOME*\bin\oidcmprec	Compare one Oracle Internet Directory with another, detect conflicts or discrepancies, and optionally resolve them. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidcred	UNIX: *ORACLE_HOME*/bin/oidcred Windows: *ORACLE_HOME*\ bin\oidcred	Add, update, or delete a credential that has been created in the Credential Store Framework. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidctl	UNIX: *ORACLE_HOME*/bin/oidctl Windows: *ORACLE_HOME*\ bin\oidctl	Start and stop Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*

Table A–1 (Cont.) Oracle Fusion Middleware Command-Line Tools

Command	Path	Description
oiddiag	UNIX: *ORACLE_HOME*/bin/oiddiag Windows: *ORACLE_HOME*\ bin\oiddiag	Collects diagnostic information for Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidmon	UNIX: *ORACLE_HOME*/ bin/oidmon Windows: *ORACLE_HOME*\bin\oidmon	Monitor Oracle Internet Directory processes. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidpasswd	UNIX: *ORACLE_HOME*/ bin/oidpasswd Windows: *ORACLE_HOME*\bin\oidpasswd	Change the Oracle Internet Directory password and otherwise restricts access for Oracle Internet Directory **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidprovtool	UNIX: *ORACLE_HOME*/bin/oidprovtool Windows: *ORACLE_HOME*\bin\oidprovtool.bat	Administer provisioning profile entries in Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidrealm	UNIX: *ORACLE_HOME*/bin/oidrealm Windows: *ORACLE_HOME*\bin\oidrealm.bat	Create multiple realms in Oracle Internet Directory. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
oidstats	UNIX: SQL command, oidstats.sql Windows: SQL command, oidstats.sql	Analyze the various database ods schema objects to estimate statistics. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
opmnctl	UNIX: *ORACLE_INSTANCE*/bin/opmnctl.exe Windows: *ORACLE_INSTANCE*\bin\opmnctl.exe	Start, stop, and get status on OPMN-managed processes. **See:** *Oracle Fusion Middleware Oracle Process Manager and Notification Server Administrator's Guide*
orapki	UNIX: *ORACLE_HOME*/bin/orapki Windows: *ORACLE_HOME*\bin\orapki.bat	Manages wallets and certificates. See Appendix H.
remtool	UNIX: *ORACLE_HOME*/ldap/bin/remtool Windows: *ORACLE_HOME*\ldap\bin\remtool	Search for problems and seek to rectify them in the event of an Oracle Internet Directory replication failure. **See:** *Oracle Fusion Middleware User Reference for Oracle Identity Management*
rwbuilder	UNIX: *ORACLE_HOME*/bin/rwbuilder Windows: *ORACLE_HOME*\bin\rwbuilder	Invoke the Reports Builder. **See:** *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*
rwcgi	UNIX: *ORACLE_HOME*/bin/rwcgi Windows: *ORACLE_HOME*\bin\rwcgi	Like `rwservlet`, translate and deliver information between HTTP and the Reports Server. The `rwservlet` command is the recommended choice; `rwcgi` is maintained only for backward compatibility. **See:** *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*
rwclient	UNIX: *ORACLE_HOME*/ bin/rwclient Windows: *ORACLE_HOME*\bin\rwclient	Parse and transfer a command line to the specified (or default) Reports Server. **See:** *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*
rwconverter	UNIX: *ORACLE_HOME*/bin/rwconverter Windows: *ORACLE_HOME*\bin\rwconverter	Convert one or more report definitions or PL/SQL libraries from one storage format to another. **See:** *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*

Table A–1 (Cont.) Oracle Fusion Middleware Command-Line Tools

Command	Path	Description
rwrun	UNIX: *ORACLE_HOME*/bin/rwrun Windows: *ORACLE_HOME*\bin\rwrun	Run a report using the Oracle Reports Services in-process server. **See:** *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*
rwserver	UNIX: *ORACLE_HOME*/bin/rwserver Windows: *ORACLE_HOME*\bin\rwserver.bat	Invoke the Reports Server. **See:** *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*
ssocfg	UNIX: sso/bin/ssocfg.sh Windows: sso\bin\ssocfg.bat	Update host, port, and protocol of Oracle Single Sign-On URL. **See:** *Oracle Fusion Middleware Administrator's Guide for Oracle Single Sign-On*, Release 10.1.3.4
ssooconf.sql	UNIX: *ORACLE_HOME*/portal/admin/plsql/sso/ssooconf.sql Windows: *ORACLE_HOME*\portal\admin\plsql\sso\ssooconf.sql	Script to point Oracle Single Sign-On server to a different Oracle Internet Directory. **See:** *Oracle Fusion Middleware Administrator's Guide for Oracle Single Sign-On* Release 10.1.3.4
wlst	UNIX: *WLS_HOME*/common/bin/wlst.sh Windows: *WLS_HOME*\common\bin\wlst.cmd UNIX: *ORACLE_HOME_for_component*/common/bin/wlst.sh Windows: *ORACLE_HOME_for_component*\common\bin\wlst.cmd	WebLogic Scripting tool **See:** Section 3.5.1 and *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

B
URLs for Components

This appendix provides the URLs needed to access Oracle Fusion Middleware components.

Table B–1 shows the URLs, and the default user 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**.

Unless otherwise noted, the password for each user is the password supplied during installation or the password you assigned to the user when you either created the user or changed the user's password.

Table B–1 URLs for Components

Component	URL (with Default Port Number)	Default User and Password
Oracle B2B	http://*host*:8001/b2b	`weblogic`
Oracle Business Activity Monitoring	http://*host*:9001/oracleBAM	`weblogic`
Oracle Business Intelligence Discoverer Plus	http://*host*:7777/discoverer/plus	n/a
Oracle Business Intelligence Discoverer Portlet Provider	http://*host*:7777/discoverer/portletprovider	n/a
Oracle Business Intelligence Discoverer Viewer	http://*host*:7777/discoverer/viewer	n/a
Oracle Directory Services Manager	https://*host*:7001/odsm	The superuser, such as `cn=orcladmin`
Oracle Enterprise Manager Fusion Middleware Control	http://*host*:7001/em	`weblogic`
Oracle Forms Services	http://*host*:*http_listen_port*/forms/frmservlet	Not Applicable
Oracle HTTP Server	http://*host*:7777	Not Applicable
Oracle Portal	http://*host*:*http_listen_port*/pls/portal	`orcladmin` Use the password that you supplied during installation.

Table B–1 (Cont.) URLs for Components

Component	URL (with Default Port Number)	Default User and Password
Oracle Reports Services	http://*host*:*http_listen_port*/reports/rwservlet	`orcladmin` The default password is the same as the `weblogic` password *of the Infrastructure* instance used by Oracle Reports.
Oracle WebCenter Discussions Server	http://*host*:8890/owc_discussions	`weblogic`
Oracle WebCenter Spaces	http://*host*:8888/webcenter	`weblogic`
Oracle WebCenter Portlets	http://*host*:8889/richtextportlet/info http://*host*:8889/wsrp-tools/info http://*host*:8889/portalTools	`weblogic`
Oracle WebCenter Wiki and Blogs Server	http://*host*:8890/owc_wiki	`weblogic`
Oracle WebLogic Server Administration Console	http://*host*:7001/console	`weblogic`

C

Port Numbers

This appendix provides information about Oracle Fusion Middleware port numbers.

It contains the following topics:

- Port Numbers by Component
- Port Numbers (Sorted by Number)

C.1 Port Numbers by Component

This section provides the following information for each Oracle Fusion Middleware 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 service. 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 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 C–1 shows the default port number and the port number range for components, sorted alphabetically by component.

Table C–1 Port Numbers Sorted by Component

Component or Service	Default Port Number	Allotted Port Range
Oracle Business Activity Monitoring	9001	9000-9080
Oracle Directory Integration Platform	7005	7005-9000
Oracle Directory Services Manager	7005	7005-9000
Oracle Forms Services Managed Server	9001	9001-9100
Oracle HTTP Server non-SSL Listen Port	7777	7777-7877
Oracle HTTP Server SSL Listen Port	4443	4443-4543
Oracle Identity Federation Server Managed Server	7499	7499-9000

Table C–1 (Cont.) Port Numbers Sorted by Component

Component or Service	Default Port Number	Allotted Port Range
Oracle Internet Directory (non-SSL)	3060	3061 to 3070, 13060 to 13070
Oracle Internet Directory (SSL)	3131	3132 to 3141, 13131 to 13141
Oracle Management Agent (used by Fusion Middleware Control)	5162	5162-6162
Oracle Notification Server Local Port	6100	6100 - 6199
Oracle Notification Server Remote Port	6200	6200 - 6299
Oracle Notification Server Request Port	6003	6003 - 6099
Oracle Portal Managed Server	9001	9001-9100
Oracle Reports Managed Server	9001	9001-9100
Oracle Virtual Directory (non-SSL)	6501	6501-6510
Oracle Virtual Directory (SSL)	7501	7501-7510
Oracle Web Cache Administration Port	7786	7781-7790
Oracle Web Cache Invalidation Port	7788	7781-7790
Oracle Web Cache Listen Port	7785	7781-7790
Oracle Web Cache SSL Listen Port	7789	7781-7790
Oracle Web Cache Statistics Port	7787	7781-7790
Oracle WebCenter Discussions Server	8890	8881-8890
Oracle WebCenter Portlets	8889	8881-8890
Oracle WebCenter Spaces	8888	8881-8890
Oracle WebCenter Wiki and Blog Server	8890	8881-8890
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
Oracle WebLogic Server SSL Listen Port for Administration Server	7002	7002-9000

C.2 Port Numbers (Sorted by Number)

Table C–2 lists Oracle Fusion Middleware ports numbers and components, sorted in ascending order by port number.

Table C–2 Port Numbers Sorted by Number

Default Port Number	Component or Service
3060	Oracle Internet Directory (non-SSL)
3131	Oracle Internet Directory (SSL)
4443	Oracle HTTP Server (SSL)

Table C–2 (Cont.) Port Numbers Sorted by Number

Default Port Number	Component or Service
5162	Oracle Management Agent
5556	Oracle WebLogic Server Node Manager Port
6003	Oracle Notification Server Request Port
6100	Oracle Notification Server Local Port
6200	Oracle Notification Server Remote Port
6501	Oracle Virtual Directory (non-SSL)
7001	Oracle WebLogic Server Listen Port for Administration Server
7002	Oracle WebLogic Server SSL Listen Port for Administration Server
7005	Oracle Directory Integration Platform
7005	Oracle Directory Services Manager
7499	Oracle Identity Federation Server Managed Server
7501	Oracle Virtual Directory (SSL)
7777	Oracle HTTP Server (non-SSL)
7785	Oracle Web Cache (non-SSL)
7786	Oracle Web Cache Administration Port
7787	Oracle Web Cache Statistics Port
7788	Oracle Web Cache Invalidation Port
7789	Oracle Web Cache (SSL)
8001	Oracle WebLogic Server Listen Port for Managed Server
8888	Oracle WebCenter Spaces
8889	Oracle WebCenter Portlets
8890	Oracle WebCenter Discussions Server and Oracle WebCenter Wiki and Blog Server
9001	Oracle Business Activity Monitoring Managed Server
9001	Oracle Forms Services Managed Server
9001	Oracle Portal Managed Server
9001	Oracle Reports Managed Server

D

Metadata Repository Schemas

Oracle Fusion Middleware components store metadata in a repository. Many components require a database repository to storeschemas to support the component. This appendix provides information about those schemas.

This appendix contains the following topics:

- Metadata Repository Schema Descriptions
- Metadata Repository Schemas, Tablespaces, and Data Files

D.1 Metadata Repository Schema Descriptions

Table D–1 lists the schemas used by Oracle Fusion Middleware components, sorted alphabetically by component. Note that the schema names are prefixed by the prefix you supplied when you ran the Repository Creation Utility.

Table D–1 Metadata Schemas Created by Repository Creation Utility

Component	Schema	Description
Oracle BPEL Process Manager	MDS SOAINFRA	MDS contains process definitions and configuration information. SOAINFRA contains instance and metadata database objects for Oracle Business Activity Monitoring and Oracle BPEL Process Manager.
Oracle Business Activity Monitoring	ORABAM MDS	Contains instance and metadata database objects for Oracle Business Activity Monitoring.
Oracle B2B	SOAINFRA	Contains the design and run-time repository. The design repository has modeling metadata and profile data for an integration. These describe the behavior of the integration and sequence of steps required to execute the business process. The modeling and profile metadata is the design of the integration prior to deployment and execution. Once the integration is deployed, the run-time repository contains the metadata required to execute the integration as well as the business process instance, event instances, role instances, and other data created during execution.
Oracle Business Intelligence	BISERVER BISCHEDULER BISCORECARD BIPUBLISHER	Contains metadata for Business Intelligence Server.
Oracle Business Rules	MDS	Contains configuration information for Oracle Business Rules.

Table D–1 (Cont.) Metadata Schemas Created by Repository Creation Utility

Component	Schema	Description
Oracle Event Processing	MDS	.cqlsx files are stored in a .MAR file, which is stored in MDS.
Oracle Directory Integration Platform	ODSSM	Contains configuration data for Oracle Directory Integration Platform.
Oracle Business Intelligence Discoverer	DISCOVERER DISCOVERER_PS	Contains metadata for Discoverer Portlet Provider, portlet definitions for user portlets, and cached data obtained by running scheduled Discoverer queries. Has `RESOURCE` and `CONNECT` privileges.
Oracle Mediator	MDS SOAINFRA	Contains metadata for Oracle Mediator.
Oracle Metadata Service	MDS	Contains metadata for applications that use MDS.
Oracle Identity Federation	OIF	Contains metadata for Oracle Identity Federation.
Oracle Internet Directory	ODS	For internal use.
Oracle Identity Manager	OIM	Contains metadata for applications that use Oracle Identity Manager.
Oracle Content Server	OCSERVER	Contains information for WebCenter Content Server.
Oracle Portal	PORTAL PORTAL_DEMO PORTAL_APP PORTAL_PUBLIC PORTAL_APPROVAL	Contains Oracle Portal database objects and code.
Oracle Single Sign-On	ORASSO	For internal use.
Oracle User Messaging	ORASDPM	Contains metadata related to User Messaging.
Oracle WebCenter Discussions	DISCUSSIONS	Contains information for WebCenter Discussion.
Oracle WebCenter Spaces	WEBCENTER MDS	Contains information for WebCenter Services Links, Lists, Tags, and Events.
SOA Infrastructure	SOAINFRA	Contains metadata related to Oracle B2B, Oracle BPEL Process Manager, Workflow, Sensor, Mediator, and CEP.
Oracle WebCenter Portlets	PORTLET	Contains information for WebCenter Portlet Producers.
Oracle WebCenter Wiki and Blog Server	WIKI	Contains information for WebCenter Wiki and Blogs Server.
Oracle Web Services Manager	MDS	Contains configuration information.

D.2 Metadata Repository Schemas, Tablespaces, and Data Files

Table D–2 lists the tablespace and default data file for each Metadata Repository schema. It is sorted alphabetically by schema name. Note that the default data files are prefixed by the prefix you assigned the schemas in RCU.

In addition to the tablespaces listed, the tablespace IAS_TEMP is always created when you create a schema with RCU. Its data file is iastemp.dbf.

Table D–2 Metadata Repository Tablespaces and Data Files

Schema	Tablespace	Default Data File
BIPUBLISHER	BIPUBLISHER	BIPUBLISHER.dbf
BISCHEDULER	BISCHEDULER	bischeduler.dbf
BISCORECARD	BISCORECARD	biscorecard.dbf
BISERVER	BISERVER	biserver.dbf
DISCOVERER	DISCO_PTM5_META	discoptm5meta.dbf
	DISCO_PTMS_CACHE	discoptm5cache.dbf
	DISCO_PSTORE	discopstore.dbf
DISCUSSIONS	IAS_DISCUSSIONS	iasjive.dbf
IAU	IAS_IAU	ias_iau.dbf
OCSERVER	OCSERVER	ocserver.dbf
	OCSERVER_TEMP	ocservertemp.dbf
IPM	IPM	ipm.dbf
MDS	MDS	iasmds.dbf
NS	IAS_NS	iasns.dbf
	IAS_NS_AQ	iasnsaq.dbf
ODI_MASTER	ODI_MASTER	odi_master_01.dbf
ODI_EXEC_WORK	ODI_EXEC_WORK	odi_exec_work_01.dbf
ODI_DEV_WORK	ODI_DEV_WORK	odi_dev_work_01.dbf
ODS	OLTS_DEFAULT	default1_oid.dbf
OIM	OIM	oim.dbf
	OIM_TEMP	oimtemp.dbf
ORABAM	ORABAM	orabam.dbf
ORASDPLS	IAS_ORASDPLS	orasdpls.dbf
ORASDPM	IAS_ORASDPM	iassdpm.dbf
	IAS_ORASDPM_AQ	iassdpmaq.dbf
ORASDPSDS	IAS_ORASDPSDS	orasdpsds.dbf
ORASDPSXDMS	IAS_ORASDPSXDMS	orasdpsxdms.dbf
ORASSO	IAS_ORASSO	iasorasso.dbf
PORTAL	PORTAL	portal.dbf
	PORTAL_IDX	portalidx.dbf
	PORTAL_LOG	portallog.dbf
	PORTAL_DOC	portaldoc.dbf
PORTLET	IAS_PORTLET	webcenter_portlet.dbf
SOAINFRA	SOAINFRA	soainfra.dbf
WEBCENTER	IAS_WEBCENTER	iaswebcenter.dbf
WIKI	IAS_WIKI	iaswiki.dbf

E

Using Oracle Fusion Middleware Accessibility Options

This appendix includes information about using Oracle Fusion Middleware accessibility options. It includes:

- 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://java.sun.com/javase/technologies/accessibility/accessbridge/`

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 **Setup,** then **My Preferences,** then **Accessibility.**

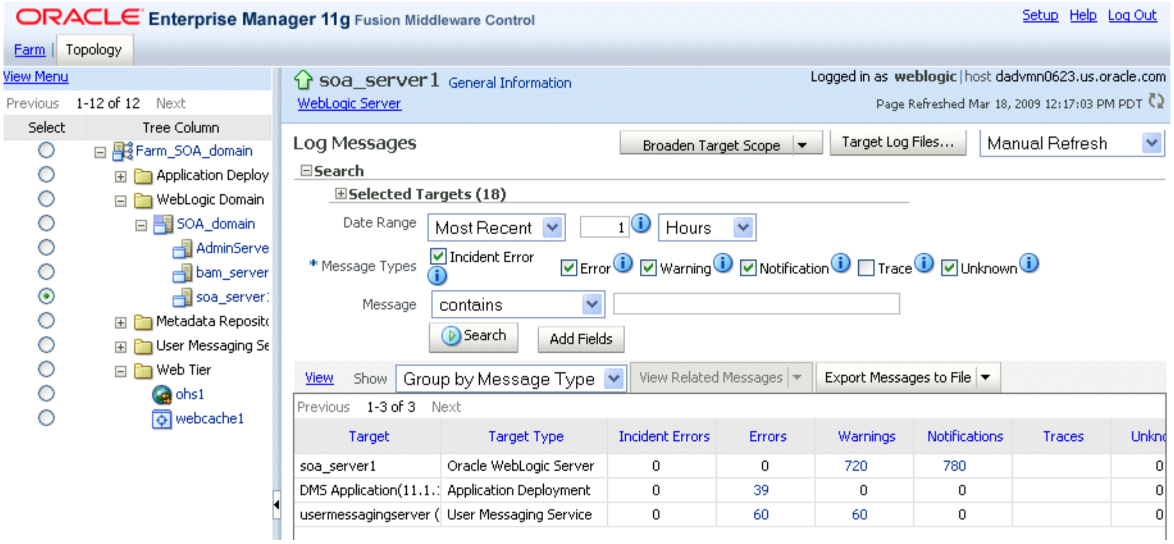
   The Accessibility Preference page is displayed.

2. Select one or both of the following options:

   - **I use a screen reader:** Accessibility-specific constructs are added to improve behavior with a screen reader.
   - **Show me the Accessibility Preference dialog when I log in:** When you log in, the Accessibility Preference dialog is displayed, with the following options:
     - I use a screen reader
     - Do not show me these options again

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:



Logged in as **weblogic** | host dadvmn0623.us.oracle.com

Page Refreshed Mar 18, 2009 12:17:03 PM PDT

Manual Refresh

Date Range: Most Recent 1 Hours

* Message Types: ☑ Incident Error ☑ Error ☑ Warning ☑ Notification □ Trace ☑ Unknown

Message: contains

Show: Group by Message Type

Target	Target Type	Incident Errors	Errors	Warnings	Notifications	Traces	Unkn
soa_server1	Oracle WebLogic Server	0	0	720	780		0
DMS Application(11.1.:	Application Deployment	0	39	0	0		0
usermessagingserver (	User Messaging Service	0	60	60	0		0

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
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

Table E–1 (Cont.) Keyboard Navigation for Common Tasks

Task	Navigation
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

F

Examples of Administrative Changes

This appendix provides examples of administrative changes that can be performed on an Oracle Fusion Middleware environment. It is a companion to Part VII, "Advanced Administration: Backup and Recovery" in this book, and to the Disaster Recovery section in *Oracle Fusion Middleware High Availability Guide*.

It contains the following topics:

- How to Use This Appendix
- Examples of Administrative Changes (by Component)

F.1 How to Use This Appendix

Some administrative operations cause configuration changes to your Oracle Fusion Middleware environment. These are called **administrative changes**, and include deploying and undeploying applications, adding or deleting Managed Servers or components, changing ports, creating and deleting users, and changing passwords. As an administrator, you should be aware when administrative changes occur because you may need to back up your environment or perform some synchronization procedures.

This appendix provides examples of administrative changes, listed by component. You can use this as a guide for performing the following procedures:

- Backup and Recovery

  Oracle recommends you perform a backup after each administrative change to your environment. You can use this appendix to determine the types of administrative changes that require you to back up your environment.

  > **See Also:** Part VII, "Advanced Administration: Backup and Recovery"

- Disaster Recovery Synchronization Between the Primary and Standby Sites

  When you implement Disaster Recovery, you must update standby sites when you make an administrative change to your environment. You can use this appendix to determine the types of administrative changes that require you to update your standby sites.

  > **See Also:** *Oracle Fusion Middleware High Availability Guide*

F.2 Examples of Administrative Changes (by Component)

Table F–1 provides examples of administrative changes, by component. Consult your component documentation to learn more about these operations.

Table F–1 Examples of Administrative Changes

Component	Examples of Administrative Changes
Directory Integration and Provisioning	Directory Integration and Provisioning administrative and configuration operations, such as running the ldapsearch utility
Dynamic Monitoring Service (DMS)	Manual edits to DMS configuration files, such as `dms.conf`
Fusion Middleware Control	Domain-wide or component-specific administrative and configuration operations performed using Fusion Middleware Control, changing port numbers, deploying and undeploying applications, and operations that result in configuration file changes
Oracle HTTP Server	Oracle HTTP Server administrative and configuration operations performed using Fusion Middleware Control, such as configuring modules, such as mod_wl_ohs, and creating virtual hosts Manual edits to Oracle HTTP Server configuration files Oracle HTTP Server administrative and configuration operations, such as registering a component with a domain, using the `opmnctl` utility
Oracle Internet Directory	Oracle Internet Directory administrative and configuration operations, such as running the `oidpasswd` utility (password management), and installing and removing components
Oracle Forms Services	Oracle Forms Services administrative and configuration operations performed using Fusion Middleware Control
Oracle Portal	Oracle Portal administrative and configuration operations performed using Fusion Middleware Control Oracle Portal administrative and configuration operations using the Administration screen in the Portal User Interface Manual edits to Oracle Portal configuration files Running the `ptlconfig` script Running any Portal-specific scripts that modify the database-side configuration for Portal, for example, disabling Oracle Web Cache or changing some background job frequencies in Portal
Oracle BPEL Process Analytics	Oracle BPEL Process Analytics administrative and configuration operations performed using Fusion Middleware Control
Oracle Reports Services	Oracle Reports Services administrative and configuration operations performed using Fusion Middleware Control, such as operations on the "Reports/Configuration" page Manual edits to Oracle Reports Services configuration files When the Reports server receives a job insert or update, such as when adding a new job or moving a job from one queue to another. *Note: Oracle recommends that you perform backup and file synchronization more frequently when running Oracle Reports Services.*
Oracle Web Cache	Oracle Web Cache configuration properties performed using Fusion Middleware Control. (**Web Cache** menu, then **Administration**).
Oracle WebLogic Server Administration Console	Domain-wide or component-specific administrative and configuration operations performed using the Administration Console, such as changing passwords, deploying and undeploying applications, and operations that result in configuration file changes

G

Viewing Release Numbers and Applying Patches

This appendix describes how to view Oracle Fusion Middleware release numbers and how to use the opatch command line to apply patches to Oracle Fusion Middleware.

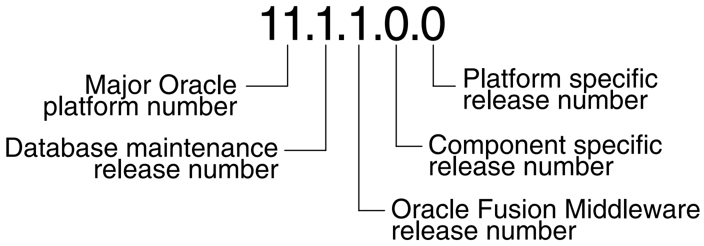
This appendix contains the following topics:

- Release Number Format
- Viewing the Software Inventory and Release Numbers
- Applying Patches

G.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 G–1.

Figure G–1 Example of an Oracle Fusion Middleware Release Number


11.1.1.0.0
Major Oracle platform number
Database maintenance release number
Platform specific release number
Component specific release number
Oracle Fusion Middleware release number

In Figure G–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.

G.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 Oracle Internet Directory Release Numbers
- Viewing Metadata Repository Release Numbers

G.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 Oracle Universal Installer, as follows:

1. Launch Oracle Universal Installer:

   ```
   (UNIX) ORACLE_HOME/oui/bin/runInstaller.sh
   (Windows) ORACLE_HOME\oui\bin\setup.exe
   ```

2. Click **Installed Products** to open the Inventory Page.
3. In the Inventory Page, expand **Oracle Homes**. You will see entries for all installations on your host.
4. Expand the Oracle home entry for the installation you are interested in.
5. You will see an entry with the release number for your original installation, followed by entries for any patch sets that have been applied.

G.2.2 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 in the following ways:

- On the File System
- Using Oracle Universal Installer

On the File System

You can view component release numbers as follows on UNIX:

```
cd ORACLE_HOME/inventory
ls -d Components*/*/*
```

Using Oracle Universal Installer

If you installed Oracle Fusion Middleware using Oracle Universal Installer, you can view component release numbers as follows:

1. Launch Oracle Universal Installer:

   ```
   (UNIX) ORACLE_HOME/oui/bin/runInstaller.sh
   (Windows) ORACLE_HOME\oui\bin\setup.exe
   ```

2. Click **Installed Products** to open the Inventory Page.
3. In the Inventory Page, expand **Oracle Homes**. You will see entries for all installations on your host.
4. Expand the Oracle home entry for the installation you are interested in.
5. You will see an entry with the release number for your original installation, followed by entries for any patch sets that have been applied.
6. Expand the initial entry to view the component release numbers at installation time. If you have subsequent patch set entries, expand them to see the component release numbers updated for each patch set.

G.2.3 Viewing Oracle Internet Directory Release Numbers

Oracle Internet Directory has a server release number, which is the version of the binaries. It also has schema and context versions. All of these numbers correspond to the Oracle Fusion Middleware installation release number through the third digit. These numbers *may* be updated when you apply a patch set release or upgrade the installation.

Viewing the Oracle Internet Directory Server Release Number

The Oracle Internet Directory server release number is the version of the binaries. You can view the Oracle Internet Directory server release number as follows:

1. Ensure that the ORACLE_HOME environment variable is set.
2. Run the following command:

   ```
   (UNIX) ORACLE_HOME/bin/oidldapd -version
   (Windows) ORACLE_HOME\bin\oidldapd -version
   ```

Viewing the Oracle Internet Directory Schema and Context Versions

You can view the Oracle Internet Directory schema and context versions in this file:

```
(UNIX) ORACLE_HOME/ldap/schema/versions.txt
(Windows) ORACLE_HOME\ldap\schema\versions.txt
```

The contents of this file are kept up-to-date, however, you can also query the schema and context release from Oracle Internet Directory, just to be sure.

To view the schema version:

1. Ensure that the ORACLE_HOME environment variable is set.
2. Run the following command:

   ```
   ldapsearch -h oid_host -p oid_port -D "cn=orcladmin"
    -q -b "cn=base,cn=oracleschemaversion"
    -s base "objectclass=*" orclproductversion
   ```

   Because you use the -q option, the command prompts you for your password.

The output will be in this form:

```
cn=BASE,cn=OracleSchemaVersion
orclproductversion=90500
```

To view the context version:

1. Ensure that the ORACLE_HOME environment variable is set.
2. Run the following command:

```
ldapsearch -h oid_host -p oid_port -D "cn=orcladmin"
 -q -b "cn=oraclecontext" -s base "objectclass=*" orclversion
```

   Because you use the -q option, the command prompts you for your password.

   The output will be in this form:

```
cn=oraclecontext
orclversion=101200
```

G.2.4 Viewing Metadata Repository Release Numbers

The Metadata Repository is an Oracle Database database that has a release number. This number is updated when you apply a patch set release or upgrade the database.

You can view the Metadata Repository release number 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                                    10.1.4.0.2      Production
Oracle Database 10g Enterprise Edition    10.1.4.0.2      64bi
PL/SQL                                    10.1.4.0.2      Production
TNS for Solaris:                          10.1.4.0.2      Production
```

G.3 Applying Patches

To apply patches, you use the following tools:

- OPatch, a utility that allows the application and rollback of interim patches to most Oracle products, such as Oracle Fusion Middleware. For the latest information about the opatch utility, and to check for updates, refer to Oracle MetaLink at

  http://www.oracle.com/support/metalink/index.html

- SmartUpdate, a standalone Java application that you can run independently of any software to upgrade Oracle WebLogic Server installations quickly and easily with maintenance patches and maintenance packs. When you install a product for the first time, Smart Update is automatically installed when you install Oracle WebLogic Server.

  For more information about using Smart Update and the products that Smart Update supports, see *Oracle Smart Update Installing Patches and Maintenance Packs*.

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.

The following sections describe how to use OPatch:

- OPatch Requirements
- Running the OPatch Utility

G.3.1 OPatch Requirements

The OPatch utility has the following requirements:

- Perl environment, included with Oracle Fusion Middleware or downloaded with a patch set.
- The Oracle home environment variable (ORACLE_HOME) must point to a valid Oracle home directory and match the value used during installation of the Oracle home directory.
- If the `-invPtrLoc` command-line argument was used during installation, then it must be used when using the OPatch utility. Oracle recommends the use of the default central inventory for a platform.
- The `jar`, `java`, `ar`, `cp`, and `make` commands must be available in the PATH statement. The commands are not available for all platforms.
- The library path must be set correctly for Oracle Real Application Clusters environments. Refer to the FAQ document in the `opatch/doc` directory for additional information.

> **See Also:** For the latest information about the OPatch utility, and to check for updates, refer to Oracle*MetaLink* at
>
> `http://www.oracle.com/support/metalink/index.html`

G.3.2 Running the OPatch Utility

The OPatch utility is located in the *ORACLE_HOME*/OPatch directory. The following shows the syntax for the OPatch utility:

```
path_to_opatch/opatch option -command_line_arguments
```

In the preceding example:

- *option*—the OPatch option. Values are described in Table G–1.
- *command_line_arguments*—the command-line arguments for each option. Values are described in the following sections.

Table G–1 Options for the OPatch Utility

Option	Description
apply	Installs an interim patch. See Section G.3.2.1.
lsinventory	Lists what is currently installed on the system. See Section G.3.2.2.
query	Queries a given patch for specific details. See Section G.3.2.3.
rollback	Removes an interim patch. See Section G.3.2.4.
version	Prints the current version of the patch tool. See Section G.3.2.5.

To view additional information for any option, use the following command:

```
path_to_OPatch/opatch option -help
```

If using Perl, then use the following command:

```
perl opatch.pl option -help
```

G.3.2.1 apply Option

The `apply` option applies an interim patch to a specified Oracle home. The ORACLE_HOME environment variable must be set to the Oracle home to be patched. The following syntax is used for this option:

```
path_to_opatch/opatch apply patch_location [-delay value] [-force] \
[-invPtrLoc path] [-jdk location] [-jre location] [-local] \
[-minimize_downtime] [-no_bug_superset] [-no_inventory] \
[-oh ORACLE_HOME_location] \
[-post options_to _be_passed_into_post [-opatch_post_end]]\
[-pre options_to_be_passed_into_pre [-opatch_pre_end]] \
[-retry value] [-silent] [-verbose]
```

The following table lists the command-line arguments for the `apply` option:

Argument	Description
delay	Specifies how many seconds to wait before attempting to lock the inventory in the case of a previous failure.
force	Removes conflicting patches from the system. If a conflict exists which prevents the patch from being applied, then the `-force` argument can be used to apply the patch.
invPtrLoc	Specifies the location of the `oraInst.loc` file. This argument is needed when the `-invPtrLoc` argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
jdk	Specifies the location of a particular JDK (jar) to use instead of the default location under the Oracle home directory.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the Oracle home directory.

Argument	Description
local	Specifies that the OPatch utility patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes. This argument can be used on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shutdown before patching, then this argument can be used for non-rolling patches.
minimize_ downtime	Specifies the order of nodes to be patched by the OPatch utility. This argument only applies to Oracle Real Application Clusters environments. It cannot be used with the `-local` argument or a rolling patch.
no_bug_superset	Specifies that the utility return an error if the current patch bugs-to-fix is a superset or the same as an installed patch bugs-fixed in the Oracle home directory.
no_inventory	Bypasses the inventory for reading and updates. This argument cannot be used with the `-local` argument. This argument puts the installation into an unsupported state.
oh	Specifies the Oracle home directory to use instead of the default.
opatch_post_end	Marks the end of the `post` options. This argument is used with the `post` argument. If this argument is not used, then everything after `post` is passed into `post`.
opatch_pre_end	Marks the end of the `pre` options. This argument is used with the `pre` argument. If this argument is not used, then everything after `pre` is passed into `pre`.
post	Specifies the parameters to be passed inside the `post` script besides the standard parameters.
pre	Specifies the parameters to be passed inside the `pre` script besides the standard parameters.
retry	Specifies how many times the OPatch utility should try when there is an inventory lock failure.
patch_location	Specifies the directory of the interim patch. This should be a directory with the same name as the patch.
silent	Suppresses user interaction, and defaults any answers to "yes."
verbose	Prints output to the screen as well as to the log file.

Note: If a patch consists of SQL changes, then they are only staged. Follow the instructions included with the patch to apply the patch manually on the affected instances. For some products, the SQL application may be implemented as a post-staging action by the tool. These patches cannot be rolled back.

G.3.2.2 lsinventory Option

The `lsinventory` option reports what has been installed on the system for a particular Oracle home directory, or for all installations. The following syntax is used for this option:

```
path_to_opatch/opatch lsinventory [-all] [-detail] [-invPrtLoc path] \
[-jre location] [-oh ORACLE_HOME_location]
```

The following table lists the command-line arguments for the `lsinventory` option:

Argument	Description
all	Reports the name and installation directory for each found Oracle home directory.
detail	Reports the installed products and other details. This argument cannot be used with the `-all` argument.
invPtrLoc	Specifies the location of the `oraInst.loc` file. This argument is needed when the `invPtrLoc` command-line argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the Oracle home directory.
oh	Specifies the Oracle home directory to use instead of the default directory.

The following is a sample output of `opatch lsinventory -detail`:

```
ORACLE_HOME      LOCATION
-----------      --------
Home1            /private/phi_local/OraHome1
   There is no Interim Patch
Home2            /private/phi_local/OraHome2
   There is no Interim Patch
Home3           /private/phi_local/OraHome6
   Installed Patch List:
    =====================
    1) Patch 20 applied on Mon Jul 11 15:53:51 PDT 2006
       [ Base Bug(s): 21  ]
    2) Patch 80 applied on Fri Jul 01 16:15:52 PDT 2006
       [ Base Bug(s): 80 81  ]
```

G.3.2.3 query Option

The `query` option queries a specific patch for specific details. It provides information about the patch and the system being patched. The following syntax is used for this option:

```
path_to_opatch/opatch query [-all] [-get_base_bug] [-get_component] \
[-invPtrLoc path] [-get_date] [-get_os] [-get_system_change] [-is_rolling] \
```

The following table lists the command-line arguments for the `query` option:

Argument	Description
all	Retrieves all information about a patch. This is equivalent to setting all command-line arguments.
get_base_bug	Describes the base bugs fixed by a patch.
get_component	Describes the Oracle components, optional or required, for a patch.
get_date	Provides the build date of a patch.
get_os	Provides the operating system description supported by a patch.
get_system_change	Describes the changes that will be made to the system by a patch. This argument is not available.

Argument	Description
invPtrLoc	Specifies the location of the `oraInst.loc` file. This argument is needed when the `invPtrLoc` command-line argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
is_rolling	Specifies if the patch is a rolling patch for Oracle Real Application Clusters. The set of patches need not be applied to the whole cluster at the same time. The patches can be applied to a select set of nodes at a time.

G.3.2.4 rollback Option

The `rollback` option removes a specific interim patch from the appropriate Oracle home directory. The following syntax is used for this option:

```
path_to_opatch/opatch rollback -id patch_id -ph patch_directory \
[-delay value] [-invPtrLoc path] [-jdk location] [-jre location]\
[-local] [-oh ORACLE_HOME_location] \
[-post options_to_be_passed_into_post [-opatch_post_end]] \
[-pre options_to_be_passed_into_pre [-opatch_pre_end]] [-retry value] \
[-silent] [-verbose]
```

The following table lists the command-line arguments for the `rollback` option:

Argument	Description
delay	Specifies how many seconds the OPatch utility should wait before attempting to lock inventory again, if the `-retry` argument is used with the `apply` option.
id	Indicates the patch to be rolled back. Use the `-lsinventory` option to display all patch identifiers. To successfully rollback a patch, the patch identifier must be supplied.
invPtrLoc	Specifies the location of the `oraInst.loc` file. This argument is needed when the `-invPtrLoc` command-line argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
jdk	Specifies the location of a particular JDK (jar) to use instead of the default location under the Oracle home directory.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the Oracle home directory.
local	Specifies that the OPatch utility patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes. This argument can be used on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shutdown before patching, then this argument can be used for non-rolling patches.
oh	Specifies the Oracle home directory to use instead of the default directory.
opatch_post_end	Marks the end of the `post` options. This argument is used with the `post` argument. If this argument is not used, then everything after `post` is passed into `post`.
opatch_pre_end	Marks the end of the `pre` options. This argument is used with the `pre` argument. If this argument is not used, then everything after `pre` is passed into `pre`.

Argument	Description
ph	Specifies the valid patch directory area. The utility will use the command types found in the patch directory to identify which commands are used for the current operating system.
post	Specifies the parameters to be passed inside the `post` script besides the standard parameters.
pre	Specifies the parameters to be passed inside the `pre` script besides the standard parameters.
retry	Specifies how many times the OPatch utility should try in case of an inventory lock failure.
silent	Suppresses user interaction, and defaults any answers to "yes."
verbose	Prints output to the screen as well as to the log file.

G.3.2.5 version Option

The `version` option shows the current version number of the OPatch utility. The following syntax is used for this option:

```
path_to_opatch/opatch version
```

H

Oracle Wallet Manager and orapki

Oracle Application Server 10*g* provided two utilities for managing wallets and certificates:

- Oracle Wallet Manager, a graphical user interface tool to manage PKI certificates
- 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

Additionally, Oracle Application Server 10*g* provided the SSL Configuration Tool.

Oracle Fusion Middleware 11*g* Release 1 (11.1.1) provides:

- Additional `orapki` features
- The ability to manage JKS-based keystores, wallets, and certificates using Fusion Middleware Control
- Both command-line and graphical user interfaces to configure SSL

Use this appendix to learn about `orapki` updates, and to help transition to the new certificate, wallet management, and SSL configuration tools provided in 11*g* Release 1 (11.1.1). The appendix contains these topics:

- New orapki Features
- Using the orapki Utility for Certificate Validation and CRL Management
- Equivalent Features for Oracle Wallet Manager
- Equivalent Features for orapki
- Equivalent Features for the SSL Configuration Tool

See Also: *Oracle Application Server Administrator's Guide* for Release 10*g* for details of Oracle Wallet Manager and `orapki` usage.

Note: The `orapki` utility is located in the binary directory of Oracle Common home.

H.1 New orapki Features

The `orapki` command-line utility contains these new features in Oracle Fusion Middleware 11*g* Release 1 (11.1.1):

- orapki Usage Examples
- New CRL Management Features

- New Version 3 Certificate Support
- Trust Chain Export
- Wallet Password Change
- Converting Between Oracle Wallet and JKS Keystore

H.1.1 orapki Usage Examples

Here are a few examples of using orapki:

```
# Create root wallet (for example, CA wallet)
orapki wallet create -wallet ./root -pwd mypasswd

# Add a self-signed certificate (CA certificate) to the root wallet
orapki wallet add -wallet ./root -dn 'CN=root_test,C=US' -keysize 1024 -self_
signed -validity 3650 -pwd mypasswd

# Export self-signed certificate from the wallet
orapki wallet export -wallet ./root -dn 'CN=root_test,C=US' -cert
./root/b64certificate.txt -pwd mypasswd

# Create a user wallet (for example, a customer wallet)
orapki wallet create -wallet ./user -pwd mypasswd

# Add a certificate request
orapki wallet add -wallet ./user -dn 'CN=user_test,C=US' -keysize 1024 -pwd
mypasswd

# Export the certificate request
orapki wallet export -wallet ./user -dn 'CN=user_test,C=US' -request
./user/creq.txt -pwd mypasswd

# Create a certificate (issued by CA)
orapki cert create -wallet ./root -request ./user/creq.txt -cert ./user/cert.txt
-validity 3650 -pwd mypasswd

# Add a trusted certificate (CA certificate) to the wallet
orapki wallet add -wallet ./user -trusted_cert -cert ./root/b64certificate.txt
-pwd mypasswd

# Add a user certificate
orapki wallet add -wallet ./user -user_cert -cert ./user/cert.txt -pwd mypasswd

# Display contents of wallet
orapki wallet display -wallet ./root -pwd mypasswd
```

H.1.2 New CRL Management Features

`orapki` supports several new command options to work with CRLs:

Creating a CRL

You use orapki crl create to create a CRL.

See Section H.2.6.3, "orapki crl create."

Revoking a Certificate

You use orapki crl revoke to revoke a certificate.

See Section H.2.6.8, "orapki crl revoke."

Verifying a CRL Signature

You use orapki crl verify to verify a CRL signature.

See Section H.2.6.11, "orapki crl verify."

Checking If a Certificate Is Revoked in a CRL

You use orapki crl status to check if a certificate is revoked.

See Section H.2.6.9, "orapki crl status."

H.1.3 New Version 3 Certificate Support

orapki provides:

- The ability to add a subject key identifier extension to a certificate request
- The ability to add a version3 self-signed certificate to a wallet

See Section H.2.6.12, "orapki wallet add" for information about these features.

H.1.4 Trust Chain Export

You use orapki wallet export_trust_chain to export a chain of trust (certificate chain) for a user.

See Section H.2.6.17, "orapki wallet export_trust_chain."

H.1.5 Wallet Password Change

You use orapki wallet change_pwd to change a wallet password.

See Section H.2.6.13, "orapki wallet change_pwd."

H.1.6 Converting Between Oracle Wallet and JKS Keystore

You can convert a JKS keystore to an Oracle wallet, and convert an Oracle wallet to JKS.

Converting JKS to Oracle Wallet

Use this command to migrate entries from JKS store to p12 wallet:

```
jks_to_pkcs12 -wallet wallet -pwd pwd -keystore keystore
-jkspwd jkspwd [-aliases [alias:alias..]]
```

where the parameters are as follows:

- `wallet` is the wallet location; entries from the JKS keystore will be migrated to this wallet.
- `pwd` is the wallet password.
- `keystore` is the keystore location; this JKS will be migrated to the p12 wallet.
- `jkspwd` is the JKS password.
- `aliases` are optional. If specified, only entries corresponding to the specified alias are migrated. If not specified, all the entries are migrated.

To illustrate this command, start by creating a self-signed JKS keystore:

```
keytool -genkey -alias myalias -keyalg RSA -keysize 1024 -dname CN=root,C=US
-validity 3650 -keystore ./ewallet.jks -storetype jks -storepass password
-keypass password
```

Next, create an Oracle wallet:

```
orapki wallet create -wallet ./ -pwd password
```

Migrate the JKS keystore entries to the wallet:

```
orapki wallet jks_to_pkcs12 -wallet ./ -pwd password -keystore ./ewallet.jks
-jkspwd password
```

> **Note:** In this example the wallet was newly created and is empty. However, in practice the wallet need not be empty when you use this command; pre-existing entries are preserved.

Converting Oracle Wallet to JKS

Use this command to migrate entries from a p12 wallet to a JKS keystore:

```
pkcs12_to_jks -wallet p12wrl -pwd p12pwd
[-jksKeyStoreLoc jksKSloc -jksKeyStorepwd jksKS_pwd]
[-jksTrustStoreLoc loc -jksTrustStorepwd pwd]
```

where the parameters are as follows:

- `wallet` is the p12 wallet location
- `pwd` is the wallet password
- `jksKeyStoreLoc` is the JKS keystore location
- `jksKeyStorepwd` is the JKS keystore password
- `jksTrustStoreLoc` is the JKS truststore location
- `jksTrustStorepwd` is the JKS truststore password

> **Note:** Passwords must have a minimum length of eight characters and contain alphabetic characters combined with numbers or special characters.

This example migrates all wallet entries to the same JKS keystore:

```
orapki wallet pkcs12_to_jks -wallet ./ -pwd mypasswd -jksKeyStoreLoc ./ewallet.jks
-jksKeyStorepwd mypasswd2
```

This example migrates keys and trusted certificate entries into separate JKS keystores:

```
orapki wallet pkcs12_to_jks -wallet ./ -pwd mypasswd
-jksKeyStoreLoc ./ewalletK.jks -jksKeyStorepwd mypasswd2
-jksTrustStoreLoc ./ewalletT.jks -jksTrustStorepwd mypasswd2
```

H.2 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

H.2.1 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
  - Uploading, listing, viewing, and deleting CRLs in Oracle Internet Directory

`orapki` allows you to import certificates in both DER and PEM formats.

H.2.1.1 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
```

H.2.1.2 Environment Setup for orapki

When running orapki, ensure that one of these following environment settings is in place:

- If running in the context of Identity Management or Web Tier or Classic installations, set `ORACLE_HOME` to point to the product installation location.
- If running in the context of Oracle SOA Suite or Oracle WebCenter installations, set `JAVA_HOME` to point to a valid JDK location that contains Java 1.5 or higher.

H.2.2 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.

H.2.3 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.

H.2.4 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.

H.2.4.1 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.

H.2.4.2 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 H.2.4.3, "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.

H.2.4.3 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`.

H.2.5 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 H.2.5.1, "About Certificate Validation with Certificate Revocation Lists"
- Section H.2.5.2, "Certificate Revocation List Management"

H.2.5.1 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.

H.2.5.1.1 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.

H.2.5.1.2 How CRL Checking Works Certificate revocation status is checked against CRLs which are located in file system directories, Oracle Internet Directory, 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 H.2.5.2.1, "Renaming CRLs with a Hash Value for Certificate Validation."

2. Oracle Internet Directory

   If the server cannot locate the CRL on the local file system and directory connection information has been configured in the `ORACLE_HOME/ldap/admin/ldap.ora` file, then the server searches in the directory. It searches the CRL subtree by using the CA's distinguished name (DN) and the DN of the CRL subtree.

   The server must have a properly configured `ldap.ora` file to search for CRLs in the directory. It cannot use the Domain Name System (DNS) discovery feature of Oracle Internet Directory. Also note that if you store CRLs in the directory, then you must use the `orapki` utility to periodically update them. See Section H.2.5.2.2, "Uploading CRLs to Oracle Internet Directory."

3. 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.

H.2.5.2 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 tasks:

- Renaming CRLs with a Hash Value for Certificate Validation
- Uploading CRLs to Oracle Internet Directory
- Listing CRLs Stored in Oracle Internet Directory
- Viewing CRLs in Oracle Internet Directory
- Deleting CRLs from Oracle Internet Directory

> **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.

You can also use LDAP command-line tools to manage CRLs in Oracle Internet Directory.

> **See Also:** Command-Line Tools Overview in the *Oracle Fusion Middleware User Reference for Oracle Identity Management* for information about LDAP command-line tools and their syntax.

H.2.5.2.1 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.

H.2.5.2.2 Uploading CRLs to Oracle Internet Directory Publishing CRLs in the directory enables CRL validation throughout your enterprise, eliminating the need for individual applications to configure their own CRLs. All applications can use the CRLs stored in the directory in which they can be centrally managed, greatly reducing the administrative overhead of CRL management and use.

The user who uploads CRLs to the directory by using `orapki` must be a member of the directory group `CRLAdmins` (`cn=CRLAdmins,cn=groups,%s_OracleContextDN%`). This is a privileged operation because these CRLs are accessible to the entire enterprise. Contact your directory administrator to be added to this administrative directory group.

To upload CRLs to the directory, enter the following at the command line:

```
orapki crl upload -crl crl_location
-ldap hostname:ssl_port -user username [-wallet wallet_location] [-summary]
```

In the preceding command, *`crl_location`* is the file name or URL in which the CRL is located, *`hostname`* and *`ssl_port`* (SSL port with no authentication) are for the system on which your directory is installed, *`username`* is the directory user who has permission to add CRLs to the CRL subtree, and *`wallet_location`* is the location of a wallet that contains the certificate of the CA that issued the CRL.

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 uploading it to the directory. Specifying the `-summary` option causes the tool to print the CRL issuer's name and the LDAP entry in which the CRL is stored in the directory.

Note:

- The `orapki` utility will prompt you for the directory password when you perform this operation.
- Ensure that you specify the directory SSL port on which the Diffie-Hellman-based SSL server is running. This is the SSL port that does not perform authentication. Neither the server authentication nor the mutual authentication SSL ports are supported by the `orapki` utility.

H.2.5.2.3 Listing CRLs Stored in Oracle Internet Directory You can display a list of all CRLs stored in the directory with `orapki`, which is useful for browsing to locate a particular CRL to view or download to your local system. This command displays the CA who issued the CRL (Issuer) and its location (DN) in the CRL subtree of your directory.

To list CRLs in Oracle Internet Directory, enter the following at the command line:

```
orapki crl list -ldap hostname:ssl_port
```

In the preceding command, the *`hostname`* and *`ssl_port`* are for the system on which your directory is installed. Note that this is the directory SSL port with no authentication as described in the preceding section.

H.2.5.2.4 Viewing CRLs in Oracle Internet Directory You can view specific CRLs that are stored in Oracle Internet Directory in a summarized format or you can request a complete listing of revoked certificates for the specified CRL. A summary listing provides the CRL issuer's name and its validity period. A complete listing provides a list of all revoked certificates contained in the CRL.

To view a summary listing of a CRL in Oracle Internet Directory, enter the following at the command line:

```
orapki crl display -crl crl_location [-wallet wallet_location] -summary
```

In the preceding command, *`crl_location`* is the location of the CRL in the directory. It is convenient to paste the CRL location from the list that displays when you use the `orapki crl list` command. See "Section H.2.5.2.3, "Listing CRLs Stored in Oracle Internet Directory".

To view a list of all revoked certificates contained in a specified CRL, which is stored in Oracle Internet Directory, enter the following at the command line:

```
orapki crl display -crl crl_location [-wallet wallet_location] -complete
```

For example, the following `orapki` command:

```
orapki crl display -crl $T_WORK/pki/wlt_crl/nzcrl.txt -wallet $T_WORK/pki/wlt_crl
-complete
```

produces the following output, which lists the CRL issuer's DN, its publication date, date of its next update, and the revoked certificates it contains:

```
issuer = CN=root,C=us, thisUpdate = Sun Nov 16 10:56:58 PST 2003,
nextUpdate = Mon Sep 30 11:56:58 PDT 2013, revokedCertificates =
{(serialNo = 153328337133459399575438325845117876415,
revocationDate - Sun Nov 16 10:56:58 PST 2003)}
CRL is valid
```

Using the `-wallet` option causes the `orapki crl display` command to validate the CRL against the CA's certificate.

Depending on the size of your CRL, choosing the `-complete` option may take a long time to display.

You can also use Oracle Directory Manager, a graphical user interface tool that is provided with Oracle Internet Directory, to view CRLs in the directory. CRLs are stored in the following directory location:

```
cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext
```

H.2.5.2.5 Deleting CRLs from Oracle Internet Directory The user who deletes CRLs from the directory by using `orapki` must be a member of the directory group `CRLAdmins`. See Section H.2.5.2.2, "Uploading CRLs to Oracle Internet Directory" for information about this directory administrative group.

To delete CRLs from the directory, enter the following at the command line:

```
orapki crl delete -issuer issuer_name -ldap hostname:ssl_port
-user username [-summary]
```

In the preceding command, *`issuer_name`* is the name of the CA who issued the CRL, the *`hostname`* and *`ssl_port`* are for the system on which your directory is installed, and *`username`* is the directory user who has permission to delete CRLs from the CRL subtree. Note that this must be a directory SSL port with no authentication. See Section H.2.5.2.2, "Uploading CRLs to Oracle Internet Directory" for more information about this port.

Using the -summary option causes the tool to print the CRL LDAP entry that was deleted.

For example, the following orapki command:

```
orapki crl delete -issuer "CN=root,C=us"
-ldap machine1:3500 -user cn=orcladmin -summary
```

produces the following output, which lists the location of the deleted CRL in the directory:

```
Deleted CRL at cn=root
cd45860c.rN,cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext
```

H.2.6 orapki Utility Commands Summary

This section lists and describes the following orapki commands:

- orapki cert create
- orapki cert display
- orapki crl create
- orapki crl delete
- orapki crl display
- orapki crl hash
- orapki crl list
- orapki crl revoke
- orapki crl status
- orapki crl upload
- orapki crl verify
- orapki wallet add
- orapki wallet change_pwd
- orapki wallet create
- orapki wallet display
- orapki wallet export
- orapki wallet export_trust_chain

H.2.6.1 orapki cert create

The following sections describe this command.

H.2.6.1.1 Purpose Use this command to create a signed certificate for testing purposes.

H.2.6.1.2 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.

H.2.6.2 orapki cert display

The following sections describe this command.

H.2.6.2.1 Purpose Use this command to display details of a specific certificate.

H.2.6.2.2 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

H.2.6.3 orapki crl create

The following sections describe this command.

H.2.6.3.1 Purpose Use this command to create a CRL.

H.2.6.3.2 Syntax
```
orapki crl create [-crl [url|filename]]
[-wallet [cawallet]]
[-nextupdate [days]]
[-pwd pwd]
```

- -crl is the location where the CRL will be created (for example `./nzcrl.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

H.2.6.4 orapki crl delete

The following sections describe this command.

H.2.6.4.1 Purpose Use this command to delete CRLs from Oracle Internet Directory. Note that the user who deletes CRLs from the directory by using `orapki` must be a member of the `CRLAdmins` (`cn=CRLAdmins,cn=groups,%s_OracleContextDN%`) directory group.

H.2.6.4.2 Syntax
```
orapki crl delete -issuer issuer_name
-ldap hostname:ssl_port -user username [-summary]
```

- The `-issuer` parameter specifies the name of the certificate authority (CA) who issued the CRL.

- The `-ldap` parameter specifies the hostname and SSL port for the directory in which the CRLs are to be deleted. Note that this must be a directory SSL port with no authentication. See Section H.2.5.2.2, "Uploading CRLs to Oracle Internet Directory" for more information about this port.
- The `-user` parameter specifies the username of the directory user who has permission to delete CRLs from the CRL subtree in the directory.
- The `-summary` parameter is optional. Using it causes the tool to print the CRL LDAP entry that was deleted.

H.2.6.5 orapki crl display

The following sections describe this command.

H.2.6.5.1 Purpose Use this command to display specific CRLs that are stored in Oracle Internet Directory.

H.2.6.5.2 Syntax `orapki crl display -crl` *`crl_location`*

```
[-wallet wallet_location] [-summary|-complete]
```

- The `-crl` parameter specifies the location of the CRL in the directory. It is convenient to paste the CRL location from the list that displays when you use the `orapki crl list` command. See Section H.2.6.7, "orapki crl list".
- 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 displaying it.
- Choosing either the -summary or the -complete parameters displays the following information:
  - `-summary` provides a listing that contains the CRL issuer's name and the CRL's validity period
  - `-complete` provides a list of all revoked certificates that the CRL contains. Note that this option may take a long time to display, depending on the size of the CRL.

H.2.6.6 orapki crl hash

The following sections describe this command.

H.2.6.6.1 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.

H.2.6.6.2 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.

H.2.6.7 orapki crl list

The following sections describe this command.

H.2.6.7.1 Purpose Use this command to display a list of CRLs stored in Oracle Internet Directory. This is useful for browsing to locate a particular CRL to view or download to your local file system.

H.2.6.7.2 Syntax `orapki crl list -ldap` *`hostname`*`:`*`ssl_port`*

The `-ldap` parameter specifies the hostname and SSL port for the directory server from which you want to list CRLs. Note that this must be a directory SSL port with no authentication. See Section H.2.5.2.2, "Uploading CRLs to Oracle Internet Directory" for more information about this port.

H.2.6.8 orapki crl revoke

The following sections describe this command.

H.2.6.8.1 Purpose Use this command to revoke a certificate.

H.2.6.8.2 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.

H.2.6.9 orapki crl status

The following sections describe this command.

H.2.6.9.1 Purpose Use this command to check if a certificate is revoked in a CRL.

H.2.6.9.2 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

H.2.6.10 orapki crl upload

The following sections describe this command.

H.2.6.10.1 Purpose Use this command to upload certificate revocation lists (CRLs) to the CRL subtree in Oracle Internet Directory. Note that you must be a member of the directory administrative group `CRLAdmins` (`cn=CRLAdmins,cn=groups,%s_OracleContextDN%`) to upload CRLs to the directory.

H.2.6.10.2 Syntax

```
orapki crl upload -crl crl_location
 -ldap hostname:ssl_port -user username
 [-wallet wallet_location] [-summary]
```

- The `-crl` parameter specifies the directory location or the URL of the CRL that you are uploading to the directory.
- The `-ldap` parameter specifies the hostname and SSL port for the directory to which you are uploading the CRLs. Note that this must be a directory SSL port with no authentication. See Section H.2.5.2.2, "Uploading CRLs to Oracle Internet Directory" for more information about this port.
- The `-user` parameter specifies the username of the directory user who has permission to add CRLs to the CRL subtree in the directory.
- The `-wallet` parameter specifies the location of the wallet that contains the certificate of the certificate authority (CA) who issued the CRL. This is an optional parameter. Using it causes the tool to verify the validity of the CRL against the CA's certificate prior to uploading it to the directory.
- The `-summary` parameter is also optional. Using it causes the tool to display the CRL issuer's name and the LDAP entry in which the CRL is stored in the directory.

H.2.6.11 orapki crl verify

The following sections describe this command.

H.2.6.11.1 Purpose Use this command to verify a CRL signature.

H.2.6.11.2 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

H.2.6.12 orapki wallet add

The following sections describe this command.

H.2.6.12.1 Purpose Use this command to add certificate requests and certificates to an Oracle wallet.

H.2.6.12.2 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 H.2.6.16, "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
```

H.2.6.13 orapki wallet change_pwd

The following sections describe this command.

H.2.6.13.1 Purpose Use this command to change the password for an Oracle wallet.

H.2.6.13.2 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.

H.2.6.14 orapki wallet create

The following sections describe this command.

H.2.6.14.1 Purpose Use this command to create an Oracle wallet or to set auto-login on for an Oracle wallet.

H.2.6.14.2 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.

H.2.6.15 orapki wallet display

The following sections describe this command.

H.2.6.15.1 Purpose Use this command to view the certificate requests, user certificates, and trusted certificates in an Oracle wallet.

H.2.6.15.2 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.

H.2.6.16 orapki wallet export

The following sections describe this command.

H.2.6.16.1 Purpose Use this command to export certificate requests and certificates from an Oracle wallet.

H.2.6.16.2 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.

H.2.6.17 orapki wallet export_trust_chain

The following sections describe this command.

H.2.6.17.1 Purpose Use this command to export a chain of trust (certificate chain) for a user.

H.2.6.17.2 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.

H.3 Equivalent Features for Oracle Wallet Manager

Table H–1 shows the wallet management features provided by Oracle Wallet Manager, and the commands or options that provide equivalent functionality in 11*g* Release 1 (11.1.1).

Table H–1 Mapping for Oracle Wallet Manager Features for Wallets

Oracle Wallet Manager Feature	How Implemented in 11gR1 Fusion Middleware Control	Notes
Creating a standard PKCS #12 wallet	**Security**, then **Wallets**	
Creating a PKCS#11 wallet	Not supported	Use Oracle Wallet Manager or the orapki command line tool
Opening a wallet	**Security**, then **Wallets**	Click on the wallet and enter a password, unless it is an auto-login wallet
Closing a wallet		Navigating to the wallets page, or opening another wallet, automatically closes the existing wallet.
Uploading a wallet to an LDAP directory	Not supported	Use the orapki command line tool
Downloading a wallet from an LDAP directory	Not supported	Use the orapki command line tool
Saving changes to an open wallet	See Notes.	Any changes made on the Manage Certificate page are automatically saved when the operation is completed.
Saving the open wallet to a new location	**Security**, then **Wallets**, then **Export**	
Saving in System Default	**Security**, then **Wallets**, then **Export**	
Deleting the wallet	**Security**, then **Wallets**, then **Delete**	
Changing the password	Not supported	Use WLST or orapki command line tools.
Enabling auto-login	See Notes.	An Auto-login wallet is automatically created with every password protected wallet.
Disabling auto-login	Not supported	You cannot disable generation of an auto-login wallet since it is always required for runtime.

Table H–2 shows the certificate management features provided by Oracle Wallet Manager, and the equivalent commands or options in 11*g* Release 1 (11.1.1).

Table H–2 Mapping for Oracle Wallet Manager Features for Certificates

Oracle Wallet Manager Feature	How Implemented in 11gR1 Fusion Middleware Control	Notes
Adding a certificate request	**Security**, then **Wallets**. Select a wallet, then **Add Certificate Request**	
Importing a user certificate	**Security**, then **Wallets**, select a wallet, then **Import**	Select User Certificate in the drop down box
Importing a trusted certificate	**Security**, then **Wallets**, select a wallet, then **Import**	Select Trusted Certificate in the drop down box
Remove certificate request	**Security**, then **Wallets**, select a wallet, select a certificate request, then **Delete**	
Remove user certificate	**Security**, then **Wallets**, select a wallet, select a user certificate, then **Delete**	
Remove trusted certificate	**Security**, then **Wallets**, select a wallet, select a trusted certificate, then **Delete**	
Export user certificate	**Security**, then **Wallets**, select a wallet, select a user certificate, then **Export**	
Export certificate request	**Security**, then **Wallets**, select a wallet, select a certificate request, then **Export**	
Export trusted certificate	**Security**, then **Wallets**, select a wallet, select a trusted certificate, then **Export**	
Export all trusted certificates	Not supported	Use WLST or `orapki` command-line tools
Importing a PKCS#7 certificate chain into the wallet	Not supported	Use WLST or `orapki` command-line tools
Exporting a PKCS#7 certificate chain from the wallet	Not supported	Use WLST or `orapki` command-line tools

Location of Default Wallet

The default location of the wallet depends on the ORACLE_HOME setting:

- When `ORACLE_HOME` is set, the default wallet location is `$ORACLE_HOME/owm/wallets/`*`username`*.
- When `ORACLE_HOME` is not set, the default wallet location is *`CurrentDir`*`/owm/wallets/`*`username`*.

H.4 Equivalent Features for orapki

Table H–3 shows the features provided by the `orapki` utility for Oracle wallets and CRLs, and the equivalent commands and options in 11*g* Release 1 (11.1.1).

Table H–3 Mapping for orapki Features for Wallets and CRLs

orapki Feature	How Implemented in 11gR1	Notes
Creating a standard PKCS#12 wallet	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

Table H–3 (Cont.) Mapping for orapki Features for Wallets and CRLs

orapki Feature	How Implemented in 11gR1	Notes
Downloading a wallet from an LDAP directory	Not supported	Use orapki command-line tool
Deleting a wallet	deleteWallet()	
Changing the wallet password	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 H–4 shows the features provided by the `orapki` utility for certificates, and the equivalent commands or options in 11*g* Release 1 (11.1.1).

Table H–4 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"

H.5 Equivalent Features for the SSL Configuration Tool

Table H–5 shows the features provided by the pre-11*g* Release 1 (11.1.1) SSL Configuration Tool, and the equivalent commands or options in 11*g* Release 1 (11.1.1).

Table H–5 Equivalent Features for the SSL Configuration Tool

SSL Configuration Tool	SSL Configuration in 11*g* Release 1 (11.1.1)
No support for wallet management	Supports management of Oracle Wallets and Java Keystores, in addition to SSL configuration
Oracle Web Cache was the only standalone type supported for SSL	Oracle HTTP Server, Oracle Web Cache, Oracle Internet Directory, and Oracle Virtual Directory are supported for standalone SSL configuration

Table H–5 (Cont.) Equivalent Features for the SSL Configuration Tool

SSL Configuration Tool	SSL Configuration in 11*g* Release 1 (11.1.1)
Provided only command line interface	Provides both command line interface (WLST) and graphical interface (Fusion Middleware Control)
Configuration file was required to run this tool. If the file was not provided, the tool prompted for values.	Configuration file is optional in the WLST command. If not provided, default values are used for SSL attributes.
Supported SSL configuration for Web tier only.	Supports SSL configuration for both Web tier and data tier.
Tool had to be run on the same physical host where component was installed.	Allows remote management of components.

I

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 topics:

- Diagnosing Oracle Fusion Middleware Problems
- Common Problems and Solutions
- Troubleshooting Fusion Middleware Control
- Need More Help?

I.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.

I.2 Common Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

- Using a Different Version of Spring
- ClassNotFound Errors When Starting Managed Servers

I.2.1 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 *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*.

I.2.2 ClassNotFound 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 this error if Node Manager has not been configured to use the start scripts when starting Managed Servers. See Section 4.2.4 for information about resolving this problem.

I.3 Troubleshooting Fusion Middleware Control

The following sections describe problems and issues when using Fusion Middleware Control:

- Troubleshooting the Display of Performance Metrics and Charts in Fusion Middleware Control
- Securing the Connection from Fusion Middleware Control to Oracle WebLogic Server Administration Console

I.3.1 Troubleshooting the Display of Performance Metrics and Charts in Fusion Middleware Control

If you are using Fusion Middleware Control to manage system components, then you might encounter situations where the performance metrics and charts do not display properly for certain managed targets.

The following sections provide information about managed targets and describe some common troubleshooting tasks to perform if Fusion Middleware Control displays errors when attempting to display performance metrics, such as response time and load metrics:

- What Are Agent-Monitored Targets?
- Setting Monitoring Credentials for All Agent-Monitored Targets in a Farm
- Changing the Monitoring Credentials for a Specific Agent-Monitored Target
- Verifying or Changing the Oracle Management Agent URL

I.3.1.1 What Are Agent-Monitored Targets?

To discover and view the following components with Fusion Middleware Control, an Oracle Management Agent must be available and running:

- Oracle Reports
- Oracle Internet Directory

- Oracle Virtual Directory
- Oracle Directory Integration Platform
- Oracle Identity Federation

These components can be referred to as agent-monitored targets.

When you install and configure an Oracle Fusion Middleware environment that includes these components, a management agent, Oracle Management Agent, is also installed and running in the Oracle instance.

In contrast, Java components and some system components can be managed by Fusion Middleware Control without a management agent.

For more information about the Oracle Management Agent, refer to the Oracle Enterprise Manager documentation on the Oracle Technology Network (OTN):

`http://www.oracle.com/technology/documentation/oem.html`

I.3.1.2 Setting Monitoring Credentials for All Agent-Monitored Targets in a Farm

To make it easier to manage the monitoring credentials for all of your agent-monitored targets, you can use the Monitoring Credentials page to set the monitoring credentials for all of the agent-monitored targets in the farm:

1. From the **Farm** menu, select **Monitoring Credentials**.
2. Enter the user name and password of an Oracle WebLogic Server user account that has at least the `monitoring` level of privileges.

When you set the monitoring credentials on this page, you override all the monitoring credentials for the agent-monitored targets in the farm. However, after you set the monitoring credentials for all the targets, you can override the credentials for a specific target by using the Agent-Monitored Targets page, as described in Section I.3.1.2.

I.3.1.3 Changing the Monitoring Credentials for a Specific Agent-Monitored Target

To manage a target (an Oracle Fusion Middleware component), the Oracle Management Agent uses an Oracle WebLogic Server administration account to connect to the target. After it connects to the target, the Oracle Management Agent can gather performance metrics and send them back to the Fusion Middleware Control where they appear on monitoring pages and in performance charts.

This administration account and its password are called the monitoring credentials for an agent-monitored target.

If the monitoring credentials for a particular target are changed in Oracle WebLogic Server, then the Oracle Management Agent can no longer obtain the performance metrics. As a result, no metrics for the target will appear on the Fusion Middleware Control pages and the performance charts will not render.

To fix this problem, you can modify the monitoring credentials of the Agent-Monitored target in Fusion Middleware Control:

1. From the **Farm** menu, select **Monitoring Credentials**.

   The Monitoring Credentials page is displayed.
2. Click **Agent-Monitored Targets.**

   The Agent-Monitored Targets page is displayed.
3. Click the Configure icon for the target that you need to modify.

4. On the Configuration page, locate the monitoring credentials fields and change the credentials to match those of an Oracle WebLogic Server user account that has at least the "monitoring" level of privileges.

I.3.1.4 Verifying or Changing the Oracle Management Agent URL

If the performance metrics for all of the agent-monitored targets in the farm are unavailable, and you have verified that the monitoring credentials for the agent-monitored targets is correct, then you might have to modify the URL used by the Oracle Management Agent to communicate with Fusion Middleware Control.

This situation can occur if you have backed up your environment and restored it to another host, or if you have moved your test environment to a production environment. In either case, the host name required in the Oracle Management Agent URL must be changed before the Oracle Management Agent can once again communicate with Fusion Middleware Control.

To modify the Oracle Management Agent URL:

1. From the **Farm** menu, select **Monitoring Credentials**.

   The Monitoring Credentials page is displayed.

2. Click **Agent-Monitored Targets.**

   The Agent-Monitored Targets page is displayed.

3. Click the Configure icon for one of the agent-monitored targets listed on the page.
4. Change the Oracle Management Agent URL.

I.3.2 Securing the Connection from Fusion Middleware Control to Oracle WebLogic Server Administration Console

By default, if you access Oracle WebLogic Server Administration Console from Fusion Middleware Control, the connection is a non-SSL connection. To access the Oracle WebLogic Server Administration Console using an SSL connection, you need to access it manually using the SSL port. Alternatively, you can enable a secure Administration port.

See "Understanding Network Channels" in the *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* for information about the admin channel and how to establish a channel.

If you want to enable a secure mode of communication with the Administration Server domain and to disable all other non-secure modes, you may need to perform the following explicit steps to enable Oracle Management Agent to monitor agent-monitored targets in Fusion Middleware Control. (See Section I.3.1.1 for information about agent-monitored targets.) These steps are needed only if you are using the default self-signed certificates on the Administration Server instance or other signed certificates whose Certification Authorities (CAs) are not available in the default trust store of the JVM used by Oracle Management Agent.

In this case, take the following steps:

1. Stop the Oracle Management Agent using the following command:

   ```
   ORACLE_HOME/bin/emctl stop agent
   ```

2. Export the certificate from Oracle WebLogic Server:

   ```
   JAVA_HOME/jdk/bin/keytool -export -alias demoidentity -file /tmp/wlcert
   -keystore MW_HOME/wlserver_10.3/server/lib/DemoIdentity.jks
   ```

When prompted, enter the password.

3. Update the JDKs default trust store (*JAVA_HOME*/jre/lib/security/cacerts) with the certificate. (This is the JDK being used by Oracle Management Agent.)

   ```
   keytool -import -alias demoidentity -trustcacerts -file /temp/wlcert -keystore
   JAVA_HOME/jre/lib/security/cacerts -storepass password
   ```

   When asked if you trust this certificate, enter `yes`.

4. Start the Oracle Management Agent using the following command:

   ```
   ORACLE_HOME/bin/emctl start agent
   ```

I.4 Need More Help?

You can find more solutions on Oracle*MetaLink*, `http://metalink.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 I.4.1.

> **See Also:** *Oracle Fusion Middleware Release Notes*, available on the Oracle Technology Network:
>
> `http://www.oracle.com/technology/documentation/index.html`

I.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/rda/README_Unix.txt
(Windows) ORACLE_HOME\rda\README_Windows.txt
```

Index