Oracle® Fusion Middleware

Administering Oracle Fusion Middleware

14*c* (14.1.2.0.0)
F85477-03
May 2025

ORACLE

Oracle Fusion Middleware Administering Oracle Fusion Middleware, 14*c* (14.1.2.0.0)

F85477-03

Primary Author: Oracle Corporation

Contents

Part V Monitoring Oracle Fusion Middleware

11 Monitoring Oracle Fusion Middleware

12 Managing Log Files and Diagnostic Data

13 Diagnosing Problems

15 Changing Oracle Fusion Middleware Network Configurations

18 Recovering Your Environment

Preface

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

Audience

This guide is intended for administrators of Oracle Fusion Middleware.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware documentation set:

- *Understanding Oracle Fusion Middleware*
- *Securing Applications with Oracle Platform Security Services*
- *High Availability Guide*
- *Understanding Oracle WebLogic Server*
- *Tuning Performance*
- *Administering Oracle SOA Suite and Oracle Business Process Management Suite*
- *Administering Oracle HTTP Server*
- *Administering Web Services*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
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.

Part I
About Oracle Fusion Middleware

Before you begin working with Oracle Fusion Middleware you need to understand some basic concepts as they relate to administering Oracle Fusion Middleware

- Introduction to Oracle Fusion Middleware

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.

- What Is Oracle Fusion Middleware?
  Oracle Fusion Middleware is a collection of standards-based software products that spans a range of tools and services including Jakarta EE and developer tools, integration services, identity management, business intelligence, and collaboration.
- Oracle Fusion Middleware Components
  Oracle Fusion Middleware Oracle WebLogic Server provides a wide variety of components, such as Oracle WebLogic Server, Oracle SOA Suite, Oracle HTTP Server and so on.

What Is Oracle Fusion Middleware?

Oracle Fusion Middleware is a collection of standards-based software products that spans a range of tools and services including Jakarta EE and developer tools, integration services, identity management, business intelligence, and collaboration.

Oracle Fusion Middleware offers complete support for development, deployment, and management.

Understanding Oracle Fusion Middleware describes Oracle Fusion Middleware concepts.

Oracle Fusion Middleware Components

Oracle Fusion Middleware Oracle WebLogic Server provides a wide variety of components, such as Oracle WebLogic Server, Oracle SOA Suite, Oracle HTTP Server and so on.

Some of the components include:

- Oracle Business Process Management (BPM) Studio is a desktop IDE application that enables process developers to implement the processes modelled by business analysts. Process developers use Oracle BPM Studio to edit the Business Process Management Notation (BPMN) process models and artifacts to complete their implementation. See *Developing Business Processes with Oracle Business Process Management Studio*.
- Oracle Coherence is the leading in-memory data grid solution that enables organizations to predictably scale mission-critical applications by providing fast access to frequently used data. See *Developing Applications with Oracle Coherence*.
- Oracle Data Integrator provides a fully unified solution for building, deploying, and managing complex data warehouses or as part of data-centric architectures in a SOA or business intelligence environment. See *Administering Oracle Data Integrator*.
- Oracle Enterprise Data Quality provides a comprehensive data quality management environment that is used to understand, improve, protect and govern data quality. EDQ facilitates best practice master data management, data integration, business intelligence, and data migration initiatives. EDQ provides integrated data quality in customer

relationship management and other applications. See *Understanding Oracle Enterprise Data Quality*.

- Oracle Enterprise Scheduler enables you can define, schedule, and run jobs. A job is a unit of work done on an application's behalf. You can run different job types, including Java, PL/SQL, binary scripts, web services and EJBs distributed across the nodes in an Oracle WebLogic Server cluster. See *Developing Applications for Oracle Enterprise Scheduler*.
- Oracle Forms is a component of Oracle Fusion Middleware used to develop and deploy Forms applications. The Forms applications provide a user interface to access the Oracle Database in an efficient and tightly coupled way. See *Working with Oracle Forms*.
- Oracle Identity Management enables organizations to effectively manage user identities across all enterprise resources. The following Identity Management components are included in this release. See *Oracle Platform Security Services*, *Oracle Security Developer Toolkit*, and *Oracle Web Services Manager*.
  - Oracle Platform Security 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 (Jakarta EE) applications. See *Securing Applications with Oracle Platform Security Services*.
  - Oracle Security Developer Tools provides the cryptographic building blocks necessary for developing robust security applications, ranging from basic tasks such as digital signatures and secure messaging to more complex projects such as securely implementing a service-oriented architecture. See *Developing Applications with Oracle Security Developer Tools*.
  - Oracle Web Services Manager 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 *Administering Web Services*.
- Oracle JDeveloper and Application Development Framework (ADF)
  - Oracle JDeveloper is a cross-platform IDE for the Oracle Fusion Middleware suite of products and runs on Windows, Linux, Mac OS X, and other UNIX-based systems. See *Developing Applications with Oracle JDeveloper*.
  - Oracle Application Development Framework (Oracle ADF) is an end-to-end application framework that builds on Jakarta EE standards and open-source technologies to simplify and accelerate implementing enterprise applications. See *Overview of Oracle ADF* in *Understanding Oracle Application Development Framework*.
- Oracle Managed File Transfer (MFT) is a high performance, standards-based, end-to-end managed file gateway. It features design, deployment, and monitoring of file transfers using a lightweight web-based design-time console. See *Using Oracle Managed File Transfer*.
- Oracle Service Bus is a configuration-based, policy-driven enterprise service bus designed for SOA life cycle management. It provides foundation capabilities for service discovery and intermediation, rapid service provisioning and deployment, and governance. See *Developing Services with Oracle Service Bus*.
- Oracle SOA Suite is a complete set of service infrastructure components, in a service-oriented architecture, for designing, deploying, and managing composite applications. Oracle SOA Suite enables services to be created, managed, and orchestrated into

composite applications and business processes. See *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

Oracle SOA Suite includes the following key components:

  - Oracle BPEL Process Manager, Business Rules, Human Workflow, Mediator
  - On-Premises and Cloud SOA Adapters
  - Oracle Business Activity Monitoring
  - Oracle B2B

- Oracle User Messaging Service (UMS) provides a common service responsible for sending out messages from applications to devices or services using various protocols. It also routes incoming messages from devices or services to applications. See *Administering Oracle User Messaging Service*.
- Oracle WebCenter Suite
  - Oracle WebCenter Content is an integrated suite of applications designed for managing content. Oracle WebCenter Content contains the Oracle WebCenter Content Server, which is used to manage the content repository. Oracle WebCenter Content can help a corporation unify, manage, and leverage all types of content across the entire enterprise. See *Understanding Oracle WebCenter Content*.
  - Oracle WebCenter Portal is 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. See *Administering Oracle WebCenter Portal*.
  - Oracle WebCenter Sites is a Web Experience Management system. It helps you build desktop and mobile websites, personalize them with targeted content, gather feedback on their success, analyze visitor interactions with the website, and test changes to your website based on your visitors' preferences. See *Developing with Oracle WebCenter Sites*.
- Oracle Web Tier
  - Oracle HTTP Server is the web server component for Oracle Fusion Middleware and provides a listener for Oracle WebLogic Server and the framework for hosting static pages, dynamic pages, and applications over the web. See *Administering Oracle HTTP Server*.
- 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. Oracle WebLogic Server is an ideal foundation for building applications based on service-oriented architecture (SOA).

  See *Understanding WebLogic Server*.

Part II
Basic Administration

Basic administration tasks include using Fusion Middleware Control, the Remote Console, and WLST commands. They also include managing wiring, starting and stopping components, and managing ports.

- Getting Started Managing Oracle Fusion Middleware
- Wiring Components to Work Together
  Service providers can publish endpoint information about their services, and clients of these services to query and bind to these services. You can wire particular Oracle Fusion Middleware components together and you can change the current wiring of components.
- Starting and Stopping Oracle Fusion Middleware
  You can start and stop Oracle Fusion Middleware, including the Administration Server, Managed Servers, and components.
- Managing Ports

2

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.

- 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.
- Getting Started Using Oracle Enterprise Manager Fusion Middleware Control
  Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer your domain. It can manage an Oracle WebLogic Server domain with its Administration Server, one or more Managed Servers, clusters, the Oracle Fusion Middleware components that are installed, configured, and running in the domain, and the applications you deploy.
- Getting Started Using Oracle WebLogic Remote Console
  Oracle WebLogic Remote Console is a graphical user interface that you use to manage an Oracle WebLogic Server domain.
- 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.
- 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.
- 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 Remote Console for the first time. You can create additional administrative accounts using the WLST command line or the Oracle WebLogic Remote Console.
- Configuring Node Manager
  Node Manager allows you to perform common operations, such as starting and stopping a Managed Server, using the Remote Console or Fusion Middleware Control.
- Basic Tasks for Configuring and Managing Oracle Fusion Middleware
  There are several tasks you need to take to configure and manage a basic Oracle Fusion Middleware environment after you have installed the software.

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 Getting Started Using Oracle Enterprise Manager Fusion Middleware Control.
- Oracle WebLogic Remote Console. See Getting Started Using Oracle WebLogic Remote Console.
- The Oracle Fusion Middleware command-line tools. See Getting Started Using the Oracle WebLogic Scripting Tool (WLST).
- The Fusion Middleware Control MBean Browser. See Getting Started Using the Fusion Middleware Control MBean Browsers.

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 Remote Console are graphical user interfaces that you can use to monitor and administer your Oracle Fusion Middleware environment. You can install Fusion Middleware Control and the Remote Console when you install most Oracle Fusion Middleware components.

Note the following:

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

You can perform some tasks with either tool, but for other tasks, you can only use one of the tools. Table 2-1 lists some common tasks and the recommended tool.

Table 2-1 Comparing Fusion Middleware Control and WebLogic Remote Console

Task	Tool to Use
Create additional Managed Servers	Fusion Middleware Control
Clone Managed Servers	WebLogic Remote Console
Cluster Managed Servers	Fusion Middleware Control
Start and stop Oracle WebLogic Server	Fusion Middleware Control or WebLogic Remote Console
Add users and groups	Fusion Middleware Control or WebLogic Remote Console if using the default embedded LDAP; if using another LDAP server, use the LDAP server's tool
Start and stop components	Fusion Middleware Control
Start and stop applications	Fusion Middleware Control
View and manage log files	Fusion Middleware Control for most log files WebLogic Remote Console for the following logs: *DOMAIN_HOME*/servers/*server_name*/data/ldap/log/EmbeddedLDAP.log *DOMAIN_HOME*/servers/*server_name*/data/ldap/log/EmbeddedLDAPAccess.log

Table 2-1 (Cont.) Comparing Fusion Middleware Control and WebLogic Remote Console

Task	Tool to Use
Change ports	Fusion Middleware Control or WebLogic Remote 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
Create data sources	Fusion Middleware Control or WebLogic Remote Console
Create connection pools	Fusion Middleware Control or WebLogic Remote Console
Browse JNDI objects	Fusion Middleware Control or WebLogic Remote Console
Create JMS queues	Fusion Middleware Control or WebLogic Remote Console
Configure JMS advanced queuing	Fusion Middleware Control or WebLogic Remote Console
Configure JMS resources	Fusion Middleware Control or WebLogic Remote Console
Control resource sharing	Fusion Middleware Control
Deploy SOA Composite applications	Fusion Middleware Control
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
Deploy ADF applications	Fusion Middleware Control
Deploy Jakarta EE applications	WebLogic Remote Console or Fusion Middleware Control
Administer Oracle WebCenter Portal	Fusion Middleware Control
Deploy Oracle WebCenter Portal applications	Fusion Middleware Control
Administer Oracle WebCenter Content	Fusion Middleware Control
Deploy Oracle WebCenter Content applications	Fusion Middleware Control
Administer Oracle WebCenter Sites	Fusion Middleware Control
Deploy Oracle WebCenter Sites applications	Fusion Middleware Control
Configure and manage auditing	Fusion Middleware Control
Configure SSL	WebLogic Remote Console for Oracle WebLogic Server Fusion Middleware Control. See Configuring SSL in Oracle Fusion Middleware.
Change passwords	Fusion Middleware Control or WebLogic Remote Console

Getting Started Using Oracle Enterprise Manager Fusion Middleware Control

Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer your domain. It can manage an Oracle WebLogic Server domain

with its Administration Server, one or more Managed Servers, clusters, the Oracle Fusion Middleware components that are installed, configured, and running in the domain, and the applications you deploy.

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

Displaying Fusion Middleware Control

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

`https://hostname.domain:9002/em`

The port number is the port number of the Administration Server. By default, the port number is 9002. The port number is listed in the following file:

`DOMAIN_HOME/config/config.xml`

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

To display Fusion Middleware Control:

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

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

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

Using Fusion Middleware Control Help

At any time while using the Fusion Middleware Control, you can select **Help** from the *username* menu at the top of the page to get more information. From the Help menu, you can select the following:

- **Contents,** which lists the contents of Help.
- **Help for This Page,** which provides context-sensitive help for the current page.

- **How Do I?,** which links to tutorial information in the documentation.
- **Documentation Library,** which links to the library on the Oracle Technology Network.
- **User Forums,** which links to Discussion Forums on the Oracle Technology Network.
- **Oracle Technology Network,** which links to the Oracle Technology Network.

Navigating Within Fusion Middleware Control

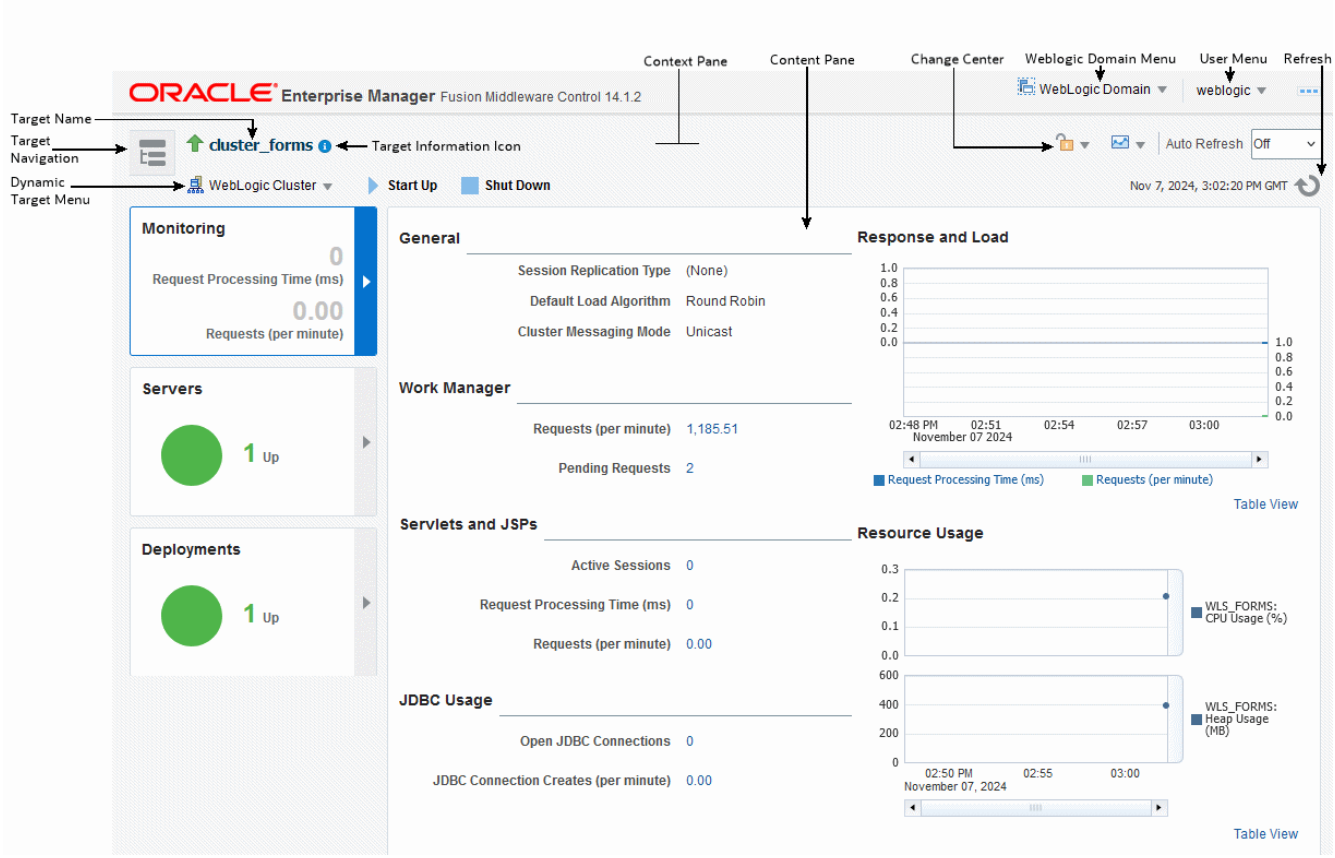
When you log into Fusion Middleware Control, it displays boxes showing the servers, clusters, and deployments on the left and the domain home page on the right.

To view the target navigation pane, click the target navigation icon near the left top corner. Fusion Middleware Control displays the target navigation pane on the left and the content pane 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 home page of an Managed Server. Because a Managed Server was selected, the dynamic target menu listed in the context pane is the WebLogic Server menu.



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

Navigating Within Fusion Middleware Control

- **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.**
- **Change Center** shows the changes made and allows you to lock and edit the configuration, release the configuration, activate changes, undo all changes, and change preferences. It also allows you to start, stop, and view your recording.
- **WebLogic Domain Menu** provides a list of operations that you can perform on the domain. The WebLogic Domain menu is always available.
- **Target Name** is the name of the currently selected target.
- **Target Information Icon** provides information about the target. For example, for a domain, it displays the target name, the version, and the domain home.
- **Target Navigation icon** expands to list all of the targets in the domain in a navigation tree.
- **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.
- **Content Pane** shows the current page for the target. When you first select a target, that target's home page is displayed.
- The ***username*** menu provides links to Help, Accessibility, information about Fusion Middleware Control, and logging out.
- **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.)
- **Right-Click Target Menu**, which you access from the target navigation pane, 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.**
- **View** lets you expand or collapse the navigation tree.

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

#unique_38/unique_38_Connect_42_CHDEHEGJ describes some common ways you can navigate within Fusion Middleware Control.

To:	Take This Action:
View all of the targets in the domain	From the View menu, select **Expand All**.
Operate on the domain	Select the **WebLogic Domain** menu, which is always available at the top right 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.

To:	Take This Action:
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 Target Information icon.
View a server log file	Right-click the server name in the **target navigation pane.** Choose **Logs,** and then **View Log Messages** to see a summary of log messages and to search log files.

About 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, as described in Table 2-2.

Table 2-2 Roles Supported by Fusion Middleware Control

Role	Actions Allowed
Administrator	All access. An administrator has full privileges, including creating and deleting instances and modifying the configuration.
Deployer	Deploy, undeploy, and redeploy applications, modify the configuration of applications, start and stop applications, create and delete JDBC and JMS resources, and modify JDBC and JMS resources, as well as all of the privileges of the Monitor role.
Operator	Start and stop servers and applications, and all of the privileges of the Monitor role.
Monitor	View configuration, status of servers and applications, metrics, log files and log messages.

Table 2-3 summarizes the privileges of each role that is supported by Fusion Middleware Control.

Table 2-3 Privileges for the Supported Roles

Privileges	Administrator	Deployer	Operator	Monitor
Edit session operations: start or release session, activate or undo changes	Yes	Yes	No	No

Table 2-3 (Cont.) Privileges for the Supported Roles

Privileges	Administrator	Deployer	Operator	Monitor
Server, Cluster, Template, or Machine				
Lifecyle operations: create, delete	Yes	No	No	No
Modify configuration	Yes	No	No	No
Control operations: start, stop, resume	Yes	No	Yes	No
View configuration	Yes	Yes	Yes	Yes
Application Deployments				
Lifecycle operations: deploy, undeploy, redeploy	Yes	Yes	No	No
Modify configuration	Yes	Yes	No	No
Control operations: start, stop	Yes	Yes	Yes	No
JDBC and JMS resources				
Lifecyle operations: create, delete	Yes	Yes	No	No
Modify configuration	Yes	Yes	No	No
Control operations: start, stop	Yes	No	No	No
View configuration	Yes	Yes	Yes	Yes
Startup and Shutdown Classes, Coherence Clusters				
Lifecyle operations: create, delete	Yes	No	No	No
Modify configuration	Yes	Yes	No	No
View configuration	Yes	Yes	Yes	Yes

Note that the information in Table 2-3 is based on the default out-of-the-box security policy for WebLogic Resources and MBeans. You can manage the default security policies in the Remote Console, as described in Security Policies and Roles in *Oracle WebLogic Remote Console Online Help*.

Understanding WebLogic Resource Security in *Securing Resources Using Roles and Policies for Oracle WebLogic Server* provides more information about resource security.

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 from Fusion Middleware Control:

1. Locate the Change Center at the top of Fusion Middleware Control.
2. From the Changes menu, select **Lock & Edit** to lock the configuration edit hierarchy for the domain.

As you make configuration changes using the Remote Console, you click **Save** (or in some cases Finish) on the appropriate pages. This does not cause the changes to take effect

immediately. The changes take effect when you click **Activate Changes** in the Change Center. At that point, the configuration changes are distributed to each of the servers in the domain. If the changes are acceptable to each of the servers, then they take effect. If any server cannot accept a change, then all of the changes are rolled back from all of the servers in the domain. The changes are left in a pending state; you can then either edit the pending changes to resolve the problem or revert to the previous configuration.

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

```
startEdit()
```

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

Viewing and Managing the WebLogic Domain

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

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

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

Viewing and Managing Java Components

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

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

1. Expand the target navigation pane, then expand **SOA.**
2. Select the SOA soa_infra instance.

   The home page for the SOA instance is displayed.
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.

Monitoring a System Component provides more information about monitoring components.

Viewing and Managing System Components

You can also view and manage system components. For example, to view and manage Oracle HTTP Server, take the following steps:

1. From the navigation pane, expand **HTTP_Server.**
2. Select the Oracle HTTP Server instance, for example, ohs1.

   The home page for the Oracle HTTP Server ohs1 is displayed.
3. From the HTTP Server menu, you can perform many administrative tasks, such as starting, stopping, and monitoring Oracle HTTP Server.

Monitoring a System Component provides more information about monitoring system components.

Getting Started Using Oracle WebLogic Remote Console

Oracle WebLogic Remote Console is a graphical user interface that you use to manage an Oracle WebLogic Server domain.

Use the Remote 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 Jakarta 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

For more information, Set Up the Console in the *Oracle WebLogic Remote Console Online Help*.

Getting Started Using the Oracle WebLogic Scripting Tool (WLST)

The Oracle WebLogic Scripting Tool (WLST) is a command-line scripting environment that you can use to create, manage, and monitor Oracle WebLogic Server domains. It is based on the Java scripting interpreter, Jython.

In addition to supporting standard Jython features such as local variables, conditional variables, and flow-control statements, WLST provides a set of scripting functions (commands) that are specific to WebLogic Server. You can extend the WebLogic scripting language to suit your needs by following the Jython language syntax

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

- Using WLST with Java Components and Oracle Fusion Middleware Services
- Using WLST Commands with System Components

Using WLST with Java Components and Oracle Fusion Middleware Services

You can use WLST commands with Java components, such as Oracle SOA Suite, Oracle Platform Security Services (OPSS), Oracle Fusion Middleware Audit Framework, and MDS, and services such as SSL, logging, and the diagnostic framework.

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

The script is located at:

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

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

```
ORACLE_HOME/oracle_common/common/bin/wlst.sh
connect('username', 'password', 'localhost:7001')
```

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

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

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

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

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

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

Introduction and Roadmap in the *WLST Command Reference for Oracle WebLogic Server* provides comprehensive information about WLST.

Using WLST Commands with System Components

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

- Oracle HTTP Server

For system components, you can only the use the WLST commands listed in Table 2-4.

Table 2-4 WLST Commands for System Components

WLST Command	Description	Additional Information
create	Creates an instance of the system component with defaults.	The create command in *WLST Command Reference for Oracle WebLogic Server*
displayLogs	Displays the messages in a log file.	The displayLogs command in *WLST Command Reference for Infrastructure Components* and Viewing Log Files and Their Messages Using WLST

Table 2-4 (Cont.) WLST Commands for System Components

WLST Command	Description	Additional Information
displayMetricTableNames	Displays the names of the DMS metric tables.	The displayMetricTableNames command in *WLST Command Reference for Infrastructure Components*
displayMetricTables	Displays the contents of the DMS metric tables.	The displayMetricTables command in *WLST Command Reference for Infrastructure Components*
dumpMetrics	Displays the available DMS metrics.	The dumpMetrics command in *WLST Command Reference for Infrastructure Components*
listLogs	Lists the log files.	The listLogs command in *WLST Command Reference for Infrastructure Components* and Viewing Log Files and Their Messages Using WLST
nmKill	Shuts down and instance.	The nmkill command in *WLST Command Reference for Oracle WebLogic Server* and Starting and Stopping System Components
nmServerStatus	Returns the status on an instance.	The nmServerStatus command in *WLST Command Reference for Oracle WebLogic Server*
nmStart	Starts an instance.	The nmStart command in *WLST Command Reference for Oracle WebLogic Server* and Starting and Stopping System Components
shutdown	Stops a system component instance.	The shutdown command in *WLST Command Reference for Oracle WebLogic Server* and Starting and Stopping System Components
start	Starts a system component instance.	The start command in *WLST Command Reference for Oracle WebLogic Server* and Starting and Stopping System Components
state	Returns the state of a system component instance.	The state command in *WLST Command Reference for Oracle WebLogic Server* and Getting Started Using the Oracle WebLogic Scripting Tool (WLST)
resync	Resynchronizes the configuration of a given system component instance in the domain.	The resync command in *WLST Command Reference for Oracle WebLogic Server*
resyncAll	Resynchronizes the configuration of all system component instances in the domain.	The resyncAll command in *WLST Command Reference for Oracle WebLogic Server*
showComponentChanges	Displays the changes made to a specified system component instance or to all system component instances in the domain.	The showComponentChanges command in *WLST Command Reference for Oracle WebLogic Server*
pullComponentChanges	Pulls the changes made to the configuration of a system component instance to the current edit session.	The pullComponentChanges command in *WLST Command Reference for Oracle WebLogic Server*

See WebLogic Server WLST Online and Offline Command Reference in the *WLST Command Reference for Oracle WebLogic Server* for information about whether a command can be invoked in online or offline mode.

To use these commands, you must invoke the WLST script from the Oracle common home. The script is located at:

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

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

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

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

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.

The following topics describe MBeans and how to view or configure MBeans:

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

About MBeans

MBeans are defined in the Jakarta EE Management Specification (the Jakarta Management Specification), 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 Jakarta EE environment. For information about the Jakarta Management Specification, see https://www.oracle.com/technetwork/es/java/javaee/tech/index.html.

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

Fusion Middleware Control provides a set of MBean browsers that allow 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 MBeans are organized into three groups: Configuration MBeans, Runtime MBeans, and Application-Defined MBeans.

Understanding WebLogic Server MBeans in *Developing Custom Management Utilities Using JMX for Oracle WebLogic Server* describes WebLogic MBeans.

Using the System MBean Browser

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

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

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

   The System MBean Browser page is displayed.
4. Expand a node in the MBean navigation tree and drill down to the MBean you want to access. Select an MBean instance.

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

   a. Click the Find icon at the top of the MBean navigation tree.

   b. For **Find,** select **MBean Name.**

      You can also select Attributes, Operations, or JMX syntax.

   c. Enter the name of the MBean and click the search icon.
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.**

The Fusion Middleware Control online help provides additional information about the MBean browser.

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 Using the System MBean Browser, and drilling down to the application. As an alternative, you can navigate to an application's MBeans using the following steps:

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

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

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 Remote Console for the first time. You can create additional administrative accounts using the WLST command line or the Oracle WebLogic Remote Console.

Understanding Users and Roles in the *Securing Applications with Oracle Platform Security Services* describes users, roles, and changing passwords.

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

- Changing the Administrative User Password Using the Command Line
- Changing the Administrative User Password Using Fusion Middleware Control

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.

MBean Reference for Oracle WebLogic Server describes the changeUserPassword method.

Changing the Administrative User Password Using Fusion Middleware Control

To change the password of an administrative user using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, select **Security,** then **Security Realm.**
2. Select the Users and Groups tab.

   The Users and Groups page is displayed.
3. Select the user.

   The Settings for *user* page is displayed.
4. Select the Passwords tab.
5. Enter the new password, then enter it again to confirm it.
6. Click **Save.**

Configuring Node Manager

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

This section describes the following topics:

- Configuring Node Manager to Start Managed Servers
- Configuring Node Manager to Use the OPSS Keystore Service

Configuring Node Manager to Start Managed Servers

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

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

```
DOMAIN_HOME/bin
```

If the Managed Servers are started by Node Manager (as is the case when the servers are started by the Remote 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_HOME/oracle_common/common/bin/setNMProps.sh.
(Windows) ORACLE_HOME\oracle_common\common\bin\setNMProps.cmd
```

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

```
ORACLE_HOME/oracle_common/common/bin/setNMProps.sh
DOMAIN_HOME/bin/startNodeManager.sh
```

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

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

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

Configuring Node Manager to Use the OPSS Keystore Service

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

1. Configure the Keystore Service, as described in How Fusion Middleware Components Use the Keystore Service in *Securing Applications with Oracle Platform Security Services*.
2. Configure Node Manager by editing the following file:

   ```
   DOMAIN_HOME/nodemanager/nodemanager.properties
   ```

   In the file, specify the following properties:

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

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

ORACLE_HOME/oracle_common/common/nodemanager/nodemanager.properties

> **Note:**
>
> Oracle Platform Security Services adds the following arguments to the startNodeManager script, which triggers the use of the Keystore Service instead of a JKS -based keystore:
>
> ```
> -Doracle.security.jps.config=DOMAIN_HOME/config/fmwconfig/jps-config-jse.xml
> -Dcommon.components.home=MW_HOME/oracle_common
> -Dopss.version=12.1.3
> ```
>
> If you configure Node Manager to start WebLogic Server without the startWebLogic script (`StartScriptEnabled=false`), you must add these arguments to the server's ServerStartMBean Arguments field using and administration tool, such as WLST or the Remote console.
>
> In addition, you must add the following to the CLASSPATH definition:
>
> *MW_HOME*/oracle_common/modules/oracle.jps_12.1.3/jps-manifest.jar.

Basic Tasks for Configuring and Managing Oracle Fusion Middleware

There are several tasks you need to take to configure and manage a basic Oracle Fusion Middleware environment after you have installed the software.

The following provides a summary of the tasks you need to take to configure and manage Oracle Fusion Middleware:

1. Configure Oracle WebLogic Server and components, such as Oracle SOA Suite or Oracle HTTP Server. See Understanding Your Installation Starting Point in *Planning an Installation of Oracle Fusion Middleware*.
2. Configure Node Manager. See Configuring Node Manager.
3. Configure SSL. See Configuring SSL in Oracle Fusion Middleware.
4. Create and manage metadata repositories, including the MDS Repository. See Creating a Database-Based Metadata Repository.
5. Deploy an application. See Deploying Applications.
6. Configure load balancing. You can configure load balancing between different components or applications. See the Server Load Balancing in a High Availability Environment in the *High Availability Guide*.
7. Back up your environment. See Introduction to Backup and Recovery.
8. Monitor your environment and manage log files. See Monitoring Oracle Fusion Middleware and Managing Log Files and Diagnostic Data.
9. Expand your environment. See Scaling Up Your Environment.

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

> **Note:**
>
> The procedures in this book for the most part assume that you are using the standard installation topology, which consists of a domain that contains an Administration Server and a cluster containing two Managed Servers.
>
> The standard topology is described in Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

3

Wiring Components to Work Together

Service providers can publish endpoint information about their services, and clients of these services to query and bind to these services. You can wire particular Oracle Fusion Middleware components together and you can change the current wiring of components.

- About Service Tables
  A **service table** provides a way for service providers to publish endpoint information about their services, and clients of these services to query and bind to these services. A service table is a single table in a database schema. There is one row for every endpoint that is published to it. The service table schema is initially created by the Repository Creation Utility.
- Viewing Service Tables
  You can view the service tables using Fusion Middleware Control.
- Wiring Components Together
  When you install and configure Oracle Fusion Middleware, most of the cross-component wiring is automatically performed. However, there may be cases when you want to connect another component or to change the current wiring.

About Service Tables

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

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

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

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

Viewing Service Tables

You can view the service tables using Fusion Middleware Control.

To view service tables:

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

   The Service Tables page is displayed.
2. You can view, edit, or delete the properties of an endpoint by clicking one of the buttons.

Wiring Components Together

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

For example:

- To connect Oracle HTTP Server to the Administration Server, so that it is connected to Fusion Middleware Control and the Remote Console. See Wiring Oracle HTTP Server to the Administration Server.
- To connect Oracle HTTP Server to an Oracle WebLogic Server cluster or server, so that applications can be routed through Oracle HTTP Server to the cluster or server. See Routing Applications Through Oracle HTTP Server to Oracle WebLogic Server.
- To connect the Oracle Web Services Manager agent to the Policy Manager. See Using Cross-Component Wiring for Auto-Discovery of Policy Manager in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

Viewing the Component End Points

You can view components' client configurations and service endpoints:

1. From the WebLogic Domain menu, select Cross-Component Wiring, then select Components.

   The Components page is displayed, showing the Client Configuration and Service End Points tables.
   If the component is part of a dynamic cluster, the URL for the connections has the format `t3s://clustername/servicename`. If the component is not part of a dynamic cluster the URL has the format `t3s://servername_or_hostname`.

2. Select a service and click **View.**

   Information about the Client Configuration or Service end point is displayed.

Wiring Oracle HTTP Server to the Administration Server

You can connect Oracle HTTP Server to the Administration Server so that you can access Fusion Middleware Control and the Administration Console through the Oracle HTTP Server, as described in the following topics:

Why Wire Oracle HTTP Server to the Administration Server?

By default, you can access Fusion Middleware Control and the WebLogic Remote Console by directly accessing the Administration Server and the default Administration port (9002). For example:

`https://hostname:9002/em`

However, in many cases, only the Oracle HTTP Server instances in the Web tier are exposed to the Internet as part of a DMZ, and the application tier (where the Administration Server resides) is protected by an additional firewall. In those cases, you can configure the Oracle HTTP Server instances in the Web tier to route any requests to the management consoles to the Administration Server. This allows administrators to access the management consoles from outside the firewall using the standard front-end URL, which is used to access the Oracle HTTP Server instances. Configuring the Web server in this way can also serve as a way of verifying the configuration of your domain, in preparation for deploying applications. When you deploy applications to the application tier, you can then configure Oracle HTTP Server in a similar manner so your application users can access the applications through the front-end HTTP Server instance URL

For a more complete example of how you might configure Oracle HTTP Server as part of a Web tier, see Configuring Oracle HTTP Server for High Availability in the *High Availability Guide*.

Connecting Oracle HTTP Server to the Administration Server

To connect Oracle HTTP Server to the Administration Server:

1. From the navigation pane, expand **HTTP Server.**
2. Select an Oracle HTTP Server instance, such as ohs1.

   The Oracle HTTP Server page is displayed.
3. From the Oracle HTTP Server menu, select **Administration,** then **mod_wl_ohs Configuration.**
4. To connect to the Administration Server, you can choose to select a cluster or a host.

   If you select **Provide WebLogic Server Host and Port Details**, provide the following details:

   - For **WebLogic Host,** enter the host name for the Administration Server.

     Alternatively, you can click the search icon. Then, select the Administration Server and click **OK.** The fields are filled in automatically.
   - For **WebLogic Port,** enter the server port for the Administration Server.

   If you select Provide WebLogic Cluster Details for **WebLogic Cluster,** enter the cluster name. Alternatively, you can click the search icon. Then, select the cluster and click **OK.** The fields are filled in automatically.

   Then, enter the following information:

   - For **Dynamic Server List ON**, check it if you want the server list to be automatically updated.
   - **Error Page**, enter the URL of a page to be shown when the server is unable to forward requests.
   - For **WebLogic Temp Directory,** enter the absolute path for a temporary directory for Oracle WebLogic Server.
   - For **Exclude Path or Mime Type**, enter paths or mime types to be excluded from being proxied.
5. For WebLogic SSL Versions, select a type.
6. In the Locations section, click **AutoFill.**

All valid WebLogic Server endpoint locations are displayed.

7. From the table, select `/em`.
8. To add the Remote Console:
   a. Click **Add Row.**
   b. For location, enter `/console`.
   c. For **WebLogic Host,** enter the host name for the Administration Server.
   d. For **Port,** enter the Administration Server port number.
9. Click **Apply.**
10. Shutdown the Oracle HTTP Server instance, then start it again.

Routing Applications Through Oracle HTTP Server to Oracle WebLogic Server

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

1. From the navigation pane, expand the domain, then **HTTP Server.**
2. Select an Oracle HTTP Server instance, such as ohs1.

   The Oracle HTTP Server page is displayed.
3. From the Oracle HTTP Server menu, select **Administration,** then **mod_wl_ohs configuration.**

   The mod_wl_ohs Configuration page is displayed.
4. You can choose to select a cluster or a host.

   If you select **Provide WebLogic Server Host and Port Details**, provide the following details:

   - For **WebLogic Host,** enter the host name for the Administration Server.

     Alternatively, you can click the search icon. Then, select the Administration Server and click **OK.** The fields are filled in automatically.
   - For **WebLogic Port,** enter the server port for the Administration Server.

   If you select Provide WebLogic Cluster Details for **WebLogic Cluster,** enter the cluster name. Alternatively, you can click the search icon. Then, select the cluster and click **OK.** The fields are filled in automatically.
5. Enter the following information:
   - For **Dynamic Server List ON**, check it if you want the server list to be automatically updated.
   - For **Error Page**, enter the URL of a page to be shown when the server is unable to forward requests.
   - For **WebLogic Temp Directory,** enter the absolute path for a temporary directory for Oracle WebLogic Server.
   - For **Exclude Path or Mime Type**, enter paths or mime types to be excluded from being proxied.
6. For WebLogic SSL Versions, select a type.
7. In the Locations section, click **AutoFill.**

All valid WebLogic Server endpoint locations are displayed.

8. Select the application from the table.
9. Click **Apply.**
10. Shutdown the Oracle HTTP Server instance, then start it again.

4

Starting and Stopping Oracle Fusion Middleware

You can start and stop Oracle Fusion Middleware, including the Administration Server, Managed Servers, and components.

- 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.
- Starting and Stopping Administration and Managed Servers and Node Manager
  You can start Oracle WebLogic Server Administration Servers using the WLST command line. You can start and stop Managed Servers using scripts, the WLST command line, the WebLogic Remote Console, or Fusion Middleware Control.
- Starting and Stopping Components
  You can start and stop components using the command line, the WebLogic Remote Console, or Fusion Middleware Control, depending upon the component.
- Starting and Stopping Fusion Middleware Control
  Fusion Middleware Control is usually automatically started or stopped when you start or stop an Oracle WebLogic Server Administration Server.
- Starting and Stopping Applications
  You can start and stop applications using Fusion Middleware Control, the WebLogic Remote Console, or the WLST command line.
- Starting and Stopping Your Oracle Fusion Middleware Environment
  An Oracle Fusion Middleware environment contains components that may be dependent on each other and should be started and stopped in a particular order.
- Starting and Stopping: Special Topics
  You may need to start and stop Oracle Fusion Middleware in a high-availability environment or to force the shutdown of a database.

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 Remote Console, or the WLST commands to start or stop Oracle Fusion Middleware components.

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

Starting and Stopping Administration and Managed Servers and Node Manager

You can start Oracle WebLogic Server Administration Servers using the WLST command line. You can start and stop Managed Servers using scripts, the WLST command line, the WebLogic Remote Console, or Fusion Middleware Control.

The following topics describe how to start and stop WebLogic Servers using the WLST command line, Fusion Middleware Control, or both:

- Starting and Stopping Administration Server
- Starting and Stopping Node Manager
- Starting and Stopping Managed Servers
- Enabling Servers to Start Without Supplying Credentials
- Setting Up Oracle WebLogic Server as a Windows Service

Starting and Stopping Administration Server

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

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

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

To stop an Administration Server, use the following script:

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

Starting and Stopping Node Manager

By default, Node Manager is configured when you configure Oracle Fusion Middleware. If Node Manager is not configured, it is very important to change the Node Manager property `StartScriptEnabled` to `True`. If this property is set to `False`, you will encounter errors or problems when starting Managed Servers configured for use by Oracle Fusion Middleware components. See Configuring Node Manager to Start Managed Servers.

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

For example, to start Node Manager, use the following script:

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

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

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

Starting and Stopping Managed Servers

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

- Starting and Stopping Managed Servers Using Fusion Middleware Control
- Starting and Stopping Managed Servers Using Scripts

Starting and Stopping Managed Servers Using Fusion Middleware Control

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

However, if the Managed Server contains other Oracle Fusion Middleware products, such as Oracle JRF or Oracle SOA Suite, you must first configure Node Manager, as described in Configuring Node Manager to Start Managed Servers.

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

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

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

Starting and Stopping Managed Servers Using Scripts

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

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

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

When prompted, enter your user name and password.

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

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

When prompted, enter your user name and password.

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

Enabling Servers to Start Without Supplying Credentials

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

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

      DOMAIN_HOME/servers/AdminServer/security

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

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

2. For each Managed Server:
   a. Create the following directory:

      DOMAIN_HOME/servers/server_name/security

   b. Copy the boot.properties file you created for the Administration Server to the security directory you created in the previous step.
3. Restart the Administration Server and Managed Servers, as described in Starting and Stopping Administration Server and Starting and Stopping Managed Servers.

> **Note:**
>
> When you start the Administration Server or Managed Server, the user name and password entries in the file are encrypted.
>
> For security reasons, minimize the time the entries in the file are left unencrypted. After you edit the file, start the server as soon as possible in order for the entries to be encrypted.

For more information about the boot.properties file, see Boot Identity Files in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

Setting Up Oracle WebLogic Server as a Windows Service

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

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

```
echo off
SETLOCAL
set DOMAIN_NAME=myWLSdomain
set USERDOMAIN_HOME=d:\Oracle\config\domains\myWLSdomain
set SERVER_NAME=myWLSserver
set PRODUCTION_MODE=true
set
JAVA_OPTIONS=-Dweblogic.Stdout="d:\Oracle\config\domains\myWLSdomain\
stdout.txt" -Dweblogic.Stderr="d:\Oracle\config\domains\myWLSdomain\stderr.txt"
set ADMIN_URL=http://adminserver:7501
set MEM_ARGS=-Xms40m -Xmx250m
```

```
call %USERDOMAIN_HOME%\bin\setDomainEnv.cmd
call "d:\Oracle_home\wlserver\server\bin\installSvc.cmd"
ENDLOCAL
```

Starting and Stopping Components

You can start and stop components using the command line, the WebLogic Remote Console, or Fusion Middleware Control, depending upon the component.

The following topics describe how to start and stop components using Fusion Middleware Control and the command line:

- Starting and Stopping Components Using Fusion Middleware Control
- Starting and Stopping Components Using the Command Line

Starting and Stopping Components Using Fusion Middleware Control

To start or stop a component:

1. From the navigation pane, navigate to the component.
2. Select the component, such as **OHS1.**
3. From the dynamic target menu, choose **Control,** then **Start Up** or **Shut Down.**

Starting and Stopping Components Using the Command Line

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

- Starting and Stopping Java Components
- Starting and Stopping System Components

Starting and Stopping Java Components

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

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

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

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

Starting and Stopping System Components

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

- To start and stop system components using scripts, use the startComponent and stopComponent scripts. You can use this method for system components, such as Oracle HTTP Server, in a standalone domain or a WebLogic Server domain. You must invoke them from the host that contains the Administration Server.

The scripts are located in

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

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

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

In the syntax:

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

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

  After you have stored the information in the properties file, when you run the scripts subsequently, you will not be prompted for a user name and password.
- `showErrorStack`: Provides more detailed error information, including all of the messages in the error stack. Specify this option if you need to determine the cause of errors.

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

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

> **Note:**
>
> You can also use these scripts to start and stop system components remotely. In that case, the scripts read the configuration to determine the location of the component.
>
> You must run these scripts from the same system as the admin server.

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

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

    ```
    nmConnect(domainName='domain_name', username='username', password='password')
    nmstart(serverName='OHS1', serverType='OHS')
    ```
  - The WLST start command. You can use this method for Oracle HTTP Server in a standalone domain.

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

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

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

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

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

To decide which method to use, note the following:

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

Starting and Stopping Fusion Middleware Control

Fusion Middleware Control is usually automatically started or stopped when you start or stop an Oracle WebLogic Server Administration Server.

If Fusion Middleware Control is configured for a domain, it is automatically started or stopped when you start or stop an Oracle WebLogic Server Administration Server, as described in Starting and Stopping Administration Server.

Starting and Stopping Applications

You can start and stop applications using Fusion Middleware Control, the WebLogic Remote Console, or the WLST command line.

The following topics describe how to start and stop applications using Fusion Middleware Control and the command line:

- Starting and Stopping Jakarta EE Applications Using Fusion Middleware Control
- Starting and Stopping Jakarta EE Applications Using WLST

Starting and Stopping Jakarta EE Applications Using Fusion Middleware Control

To start or stop a Jakarta 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.**

For information about starting and stopping Oracle Fusion Middleware components, such as Oracle SOA Suite or Oracle WebCenter Portal, see the administration guide for that component.

Starting and Stopping Jakarta EE Applications Using WLST

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

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

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

Starting and Stopping Your Oracle Fusion Middleware Environment

An Oracle Fusion Middleware environment contains components that may be dependent on each other and should be started and stopped in a particular order.

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

- Starting an Oracle Fusion Middleware Environment
- Stopping an Oracle Fusion Middleware Environment

Starting an Oracle Fusion Middleware Environment

To start an Oracle Fusion Middleware environment:

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

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

   e. Start the database instance:

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

   For more information about starting an Oracle Database, see *Oracle Database Administrator's Guide*
2. Start the Administration Server as described in Starting and Stopping Administration Server.
3. Start Node Manager as described in Starting and Stopping Node Manager.
4. Start any Oracle Identity Management components, such as Oracle Internet Directory, which form part of your environment.

5. Start the Managed Servers as described in Starting and Stopping Managed Servers Using Scripts.

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

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

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

Stopping an Oracle Fusion Middleware Environment

To stop an Oracle Fusion Middleware environment:

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

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

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

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

   e. Stop the Net Listener:

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

   For more information about stopping an Oracle Database, see the Shutting Down a Database in the *Oracle Database Administrator's Guide*

Starting and Stopping: Special Topics

You may need to start and stop Oracle Fusion Middleware in a high-availability environment or to force the shutdown of a database.

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

Starting and Stopping in High Availability Environments

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

- active-active solutions
- active-passive solutions

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

Forcing a Shutdown of an Oracle Database

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

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

An immediate database shutdown proceeds with the following conditions:

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

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

For more information about shutting down an Oracle Database, see Shutting Down a Database in the *Oracle Database Administrator's Guide* in the Oracle Database documentation library.

5

Managing Ports

You can view and change Oracle Fusion Middleware port numbers, such as those used by Oracle WebLogic Server or Oracle HTTP Server.

- About Managing Ports
  Many Oracle Fusion Middleware components and services use ports. Most port numbers are assigned during domain creation. As an administrator, it is important to know the port numbers used by these services, and to ensure that the same port number is not used by two services on your host.
- Viewing Port Numbers
  Often, you need to know which port numbers your Oracle Fusion Middleware environment uses.
- 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 Remote Console, or the command line.

About Managing Ports

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

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

See Port Numbers for a list of port numbers commonly assigned during installation. Refer to the installation guide for directions on overriding port assignments during installation.

Viewing Port Numbers

Often, you need to know which port numbers your Oracle Fusion Middleware environment uses.

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

Viewing Port Numbers Using the Command Line

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

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

Viewing Port Numbers Using Fusion Middleware Control

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

For example, to view the ports of a domain:

- From the WebLogic Domain menu, choose **Monitoring,** then **Port Usage.**

  The Port Usage page is displayed.

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

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

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

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 Remote Console, or the command line.

You can change a port number to any number, if it is an unused port. You do not have to use a port in the allotted port range for the component. See Port Numbers for information on allotted port ranges.

For information about changing other ports, see:

- Configuring Node Manager in *Administering Node Manager for Oracle WebLogic Server* for information about changing the Node Manager port.
- Changing the Oracle WebLogic Server Listen Ports
- Changing the Node Manager Listen Port
- Changing the Oracle HTTP Server Listen Ports
- Changing the Oracle Database Net Listener Port

Changing the Oracle WebLogic Server Listen Ports

You can change the non-SSL (HTTP) listen port and the SSL (HTTPS) listen port for an Administration Server or a Managed Server using Fusion Middleware Control or WLST.

See Configuring the Listen Port in *Administering Server Environments for Oracle WebLogic Server* for more information about changing Oracle WebLogic Server ports.

- Changing the Oracle WebLogic Server Listen Ports Using Fusion Middleware Control
- Changing the Oracle WebLogic Server Listen Ports Using WLST

Changing the Oracle WebLogic Server Listen Ports Using Fusion Middleware Control

To change the non-SSL (HTTP) listen port and the SSL (HTTPS) listen port for an Administration Server or a Managed Server using Fusion Middleware Control:

1. From the target navigation pane, select the server.

2. From the WebLogic Server menu, select **Administration,** then **General Settings.**
3. Select the Configuration tab. On the General Settings tab, change the number of the **Listen Port** or **SSL Listen Port.**
4. If the server is running, restart the server.
5. If other components rely on the Oracle WebLogic Server listen ports, you must reconfigure those components.

Changing the Oracle WebLogic Server Listen Ports Using WLST

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

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

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

Changing the Node Manager Listen Port

You can change the Node Manager listen port using Fusion Middleware Control or WLST, as described in the following topics:

- Changing the Node Manager Listen Port Using WLST
- Changing the Node Manager Listen Port Using Fusion Middleware Control

Changing the Node Manager Listen Port Using WLST

To change the Node Manager Listen Port using WLST:

```
readDomain('Domain_Home')
cd('/')
cd('NMProperties')
set('ListenPort',new_port_number)
updateDomain()
```

Changing the Node Manager Listen Port Using Fusion Middleware Control

To change the Node Manager Listen Port using Fusion Middleware Control, change the configuration of the machine to point it to the new port:

1. Edit the Node Manager properties file, changing the Listen Port property. For a domain-based Node Manager, the file is located at:

   DOMAIN_HOME/nodemanager/nodemanager.properties
2. From the WebLogic Domain menu, select **Environment** and then **Machines**.
3. On the Machines page, select the machine.
4. Select the Configuration tab.
5. Select the Node Manager tab.
6. Change the **Listen Port** to the new port number.

7. Click **Save.**

Changing the Oracle HTTP Server Listen Ports

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

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

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

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

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.

For information about enabling the Listen port to run as root see Starting Oracle HTTP Server Instances on a Privileged Port (Linux and UNIX Only) in *Administering Oracle HTTP Server*.

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

To change the Oracle HTTP Server non-SSL (HTTP) Listen port, take the following steps. 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 Enabling Oracle HTTP Server to Run as Root for Ports Set to Less Than 1024 (UNIX Only).

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

1. From the navigation pane, expand **HTTP Server.** 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. From the Oracle HTTP Server menu, choose **Administration,** then **Advanced Configuration.**
6. Select the related configuration file and modify the related VirtualHost directive, changing the VirtualHost's old listen port value to the new value. For example:

   ```
   <VirtualHost *:7777>
   ```

   Save the configuration.
7. Restart Oracle HTTP Server. (From the Oracle HTTP Server menu, choose **Control,** then **Restart.**)

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

To change the Oracle HTTP Server SSL (HTTPS) Listen port, take the following steps. 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 Enabling Oracle HTTP Server to Run as Root for Ports Set to Less Than 1024 (UNIX Only).

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

1. From the navigation pane, expand **HTTP Server.** 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. From the Oracle HTTP Server menu, choose **Administration,** then **Advanced Configuration.**
6. Select the related configuration file and modify the related VirtualHost directive, changing the VirtualHost's old listen port value to the new value. For example:

   ```
   <VirtualHost *:4443>
   ```

   Save the configuration.
7. Restart Oracle HTTP Server. (From the Oracle HTTP Server menu, choose **Control,** then **Restart.**)

Changing the Oracle HTTP Server Listen Ports in a Standalone Domain

To change the Oracle HTTP Server non-SSL and SSL Listen ports in a standalone domain:

1. Make a copy of the following configuration files.

   ```
   DOMAIN_HOME/config/fmwconfig/components/OHS/component_name/httpd.conf
   DOMAIN_HOME/config/fmwconfig/components/OHS/component_name/admin.conf
   DOMAIN_HOME/config/fmwconfig/components/OHS/component_name/ssl.conf
   ```
2. Modify the following configuration files and any other configuration files that hold the port values:

   ```
   DOMAIN_HOME/config/fmwconfig/components/OHS/component_name/httpd.conf
   DOMAIN_HOME/config/fmwconfig/components/OHS/component_name/ssl.conf
   ```
3. In the files, replace the non-SSL and SSL Listen ports with the new values.
4. In the files, modify the related Virtual Host ports.

Changing the Oracle Database Net Listener Port

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

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

Note that multiple Oracle Database 10*g*, Oracle Database 11*g*, Oracle Database 12*c*, Oracle Database 14*c* 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*, Oracle Database 11*g*, Oracle Database 12*c*, Oracle Database 14*c* database, they can all use port 1521. There is no need to change the listener port number.

> **Note:**
>
> To run two listeners that use the same key value on one host, refer to Changing the KEY Value for an IPC Listener

To change the database listener port:

1. Stop all components that use the metadata repository. See Starting and Stopping Oracle Fusion Middleware for instructions.
2. On the metadata repository host, change the Oracle Net listener port for the metadata repository:
   a. Ensure that the ORACLE_HOME and ORACLE_SID environment variables are set.
   b. Stop the metadata repository listener:

      ```
      lsnrctl stop
      ```

   c. 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.

   d. 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:

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

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

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

   e. Start the metadata repository listener:

      ```
      lsnrctl start
      ```

   f. 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;
      ```

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

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

3. Change the system data source to use the new port number for the metadata repository. To do so, you can use Fusion Middleware Control:
   a. In the Change Center, click **Lock & Edit.**
   b. In the navigation pane, expand select the domain.

      The WebLogic Domain page is displayed.
   c. From the WebLogic Domain menu, select **JDBC Data Sources.**

      The Summary of JDBC Data Sources page is displayed.
   d. Select the data source you want to change.

      The JDBC Data Source page is displayed.
   e. Select the Connection Pool tab.
   f. To change the database port, modify the **Database URL** field. For example:

      ```
      jdbc:oracle:thin:@hostname.domainname.com:1522/orcl
      ```
   g. Click **Save.**
   h. Restart the servers that use this data source. (Click the Targets tab to see the servers that use this data source.)

- Changing the KEY Value for an IPC Listener

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 Communication

Oracle Fusion Middleware uses technologies such as Secure Sockets Layer (SSL), keystores, wallets, and certificates; and standards such as FIPS to secure communication between its components.

- Configuring SSL in Oracle Fusion Middleware
- Managing Keystores, Wallets, and Certificates
- FIPS 140 Support in Oracle Fusion Middleware

6

Configuring SSL in Oracle Fusion Middleware

Secure Sockets Layer (SSL) is used to secure communications between Oracle Fusion Middleware components in web, middle, and data tiers. This chapter provides procedures to secure communications, describes advanced scenarios beyond the basic topologies and explains best practices.

> **Note:**
>
> The term "SSL" is used generically in this document to denote secure transport mechanisms including Transport Layer Security (TLS).
>
> Refer to component-specific documentation to learn about the cipher suite(s) that the client can use during the SSL/TLS handshake.

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

- How SSL Works
  Secure Sockets Layer (SSL) provides a secured communication between two entities, that is, a client and a server over an unsecured network. The SSL communication is achieved by using a set of protocols in both the handshake and data transfer phases. You can also review the basic concepts of SSL, cryptography in general, Oracle Wallet and keystore.
- About SSL in Oracle Fusion Middleware
  You can review the keystores and wallets that are central to SSL communication and other features like TLS cryptographic libraries that Oracle Fusion Middleware support.
- Configuring SSL for Configuration Tools
  Several tools are available for Oracle Fusion Middleware configuration. These tools must be configured with SSL to enable them to connect to an SSL-enabled Oracle WebLogic Server.
- Configuring SSL for the Web Tier
  Oracle HTTP Server resides in the web tier of the Oracle Fusion Middleware. It secures communications by using a SSL protocol.
- Configuring SSL for the Middle Tier
  SSL in the middle tier of the Oracle Fusion Middleware includes enabling SSL for the application server and also the components and applications running on the application server.
- Configuring SSL for the Data Tier
  The data tier of the Oracle Fusion Middleware includes Oracle Database as a component. All components in the data tier must be SSL enabled.

- Setting Up One-Way SSL to the LDAP Security Store
  Follow these steps to set up a one-way Secure Sockets Layer (SSL) channel between Oracle WebLogic Server or a Java SE application, and the LDAP security store. Such connection may be required, for example, when reassociating to an LDAP target store.
- Setting Up SSL in Identity Store Services
  Establishing SSL connections in Identity Store Services using libOVD and JKS requires keystore certificates that are kept in multiple locations, such as the WebLogic Server truststore and the `adapters.jks` file.
- Advanced SSL Scenarios
  SSL configuration in Oracle Fusion Middleware also includes additional scenarios that are beyond the basic topologies like FIPS 140 support, Hardware Security Modules and certificate validation.
- Best Practices for SSL
  Oracle Fusion Middleware recommends some best practices for component administrators and application developers while configuring SSL.
- WLST Reference for SSL
  WLST commands are available to manage Oracle wallets and to configure SSL for Oracle Fusion Middleware components.

How SSL Works

Secure Sockets Layer (SSL) provides a secured communication between two entities, that is, a client and a server over an unsecured network. The SSL communication is achieved by using a set of protocols in both the handshake and data transfer phases. You can also review the basic concepts of SSL, cryptography in general, Oracle Wallet and keystore.

- 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.
- About Private and Public Key Cryptography
  SSL uses both private and public key cryptography to provide message integrity, authentication, and encryption.
- Keystores and Wallets
  In Oracle Fusion Middleware, components such as Oracle HTTP Server 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.
- 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.

What SSL Provides

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

Encryption provides confidentiality by allowing only the intended recipient to read the message. SSL can use different encryption algorithms to encrypt messages. During the SSL handshake

that occurs at the start of each SSL session, the client and the server negotiate which algorithm to use. Examples of encryption algorithms supported by SSL include AES, RC4, and 3DES.

Integrity ensures that a message sent by a client is received intact by the server, untampered. To ensure message integrity, the client hashes the message into a digest using a hash function and sends this message digest to the server. The server also hashes the message into a digest and compares the digests. Because SSL uses hash functions that make it computationally infeasible to produce the same digest from two different messages, the server can tell that if the digests do not match, someone had tampered with the message. An example of a hash function supported by SSL is SHA2.

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. 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.

About Private and Public Key Cryptography

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

Secret Key Cryptography

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

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

Public Key Cryptography

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

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

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

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

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

Keystores and Wallets

In Oracle Fusion Middleware, components such as Oracle HTTP Server use the Oracle Wallet as their storage mechanism. An Oracle wallet is a container that stores your credentials, such as certificates, trusted certificates, certificate requests, and private keys.

You can store Oracle wallets on the file system or in LDAP directories such as Oracle Internet Directory. Oracle wallets can be auto-login or password-protected wallets.

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

> **Note:**
>
> You can take advantage of the central storage and unified management available with the Keystore Service to manage wallets and their contents through the export, import, and synchronization features of that service. For details about the `importKeyStore`, `exportKeyStore`, and `syncKeyStore` commands, see “Infrastructure Security Custom WLST Commands” in the *WLST Command Reference for Infrastructure Security*.

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

For more information about configuring keystores and wallets, see:

- About SSL in Oracle Fusion Middleware for a fuller description of keystore and wallet usage in Oracle Fusion Middleware
- Managing Keystores, Wallets, and Certificates for a discussion of these terms, and administration details

How SSL Sessions Are Conducted

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

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

Figure 6-1 shows the handshake messages for a typical SSL connection between a Web server and a browser. The following steps are shown in the figure:

1. The client sends a Hello message to the server.

   The message includes a list of algorithms supported by the client and a random number that will be used to generate the keys.

2. The server responds by sending a Hello message to the client. This message includes:

   - The algorithm to use. The server selected this from the list sent by the client.
   - A random number, which will be used to generate the keys.

3. The server sends its certificate to the client.

4. The client authenticates the server by checking the validity of the server's certificate, the issuer CA, and optionally, by checking that the host name of the server matches the subject DN. The client sends a Session ID for session caching.

5. The client generates a random value ("pre-master secret"), encrypts it using the server's public key, and sends it to the server.

6. The server uses its private key to decrypt the message to retrieve the pre-master secret.

7. The client and server separately calculate the keys that will be used in the SSL session.

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

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

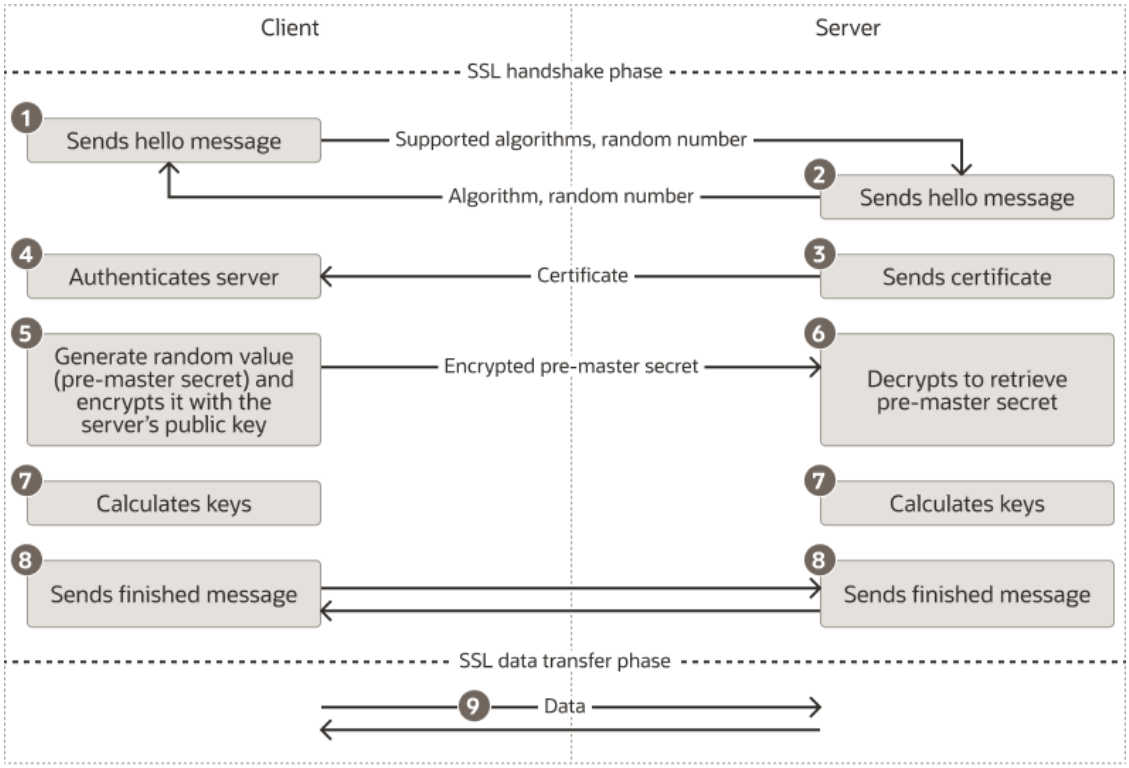
   The encryption keys are symmetric, that is, the same key is used to encrypt and decrypt the data.

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

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

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

Figure 6-1 SSL Handshake



About SSL in Oracle Fusion Middleware

You can review the keystores and wallets that are central to SSL communication and other features like TLS cryptographic libraries that Oracle Fusion Middleware support.

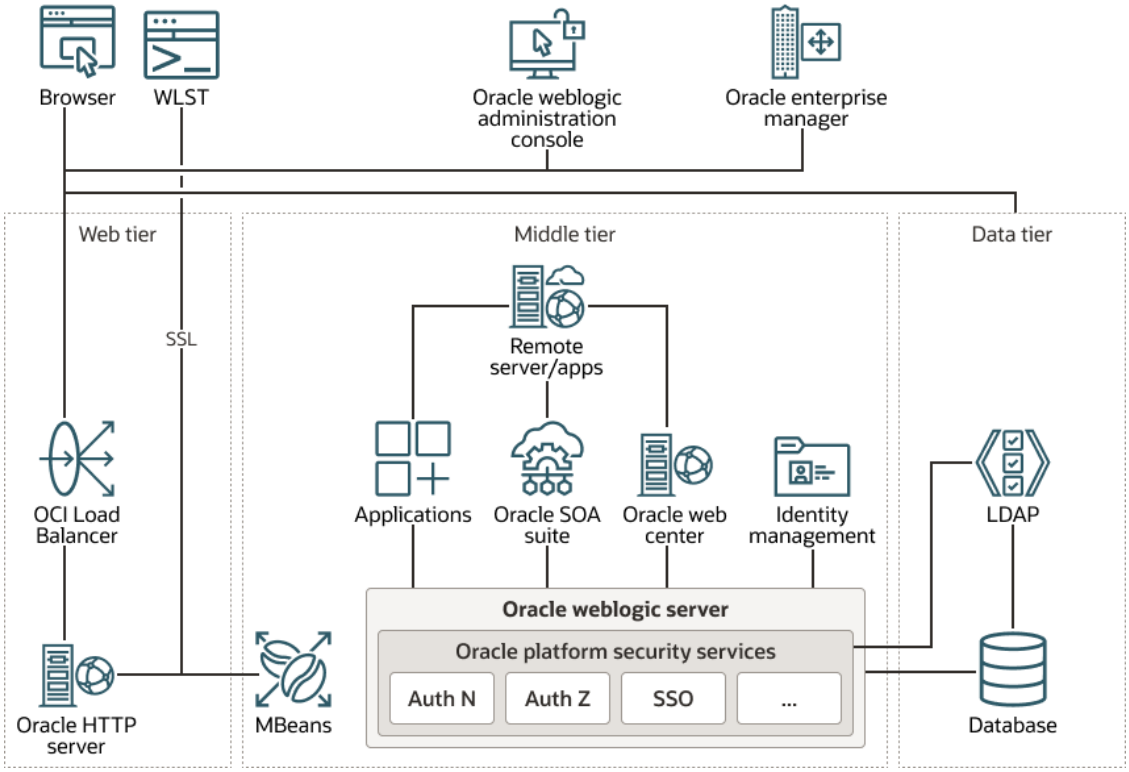
This section introduces SSL in Oracle Fusion Middleware. It contains these topics:

- SSL in the Oracle Fusion Middleware Architecture
  Review this topic for the role of SSL communication among the various components of the Oracle Fusion Middleware in its architecture.
- Keystores and Oracle Wallets
  Oracle Fusion Middleware 14*c* (14.1.2.0.0) supports different types of keystores for keys and certificates.
- TLS Protocol Support in Oracle Fusion Middleware
  Oracle Fusion Middleware supports TLS v1.2 cryptographic libraries. However, you will need to configure this protocol in the component since the library may not be enabled by default.
- Authentication Modes in Oracle Fusion Middleware
  Review this topic for the different modes of authentication that are supported between a client and a server in Oracle Fusion Middleware.
- Tools for SSL Configuration
  Oracle Fusion Middleware uses two kinds of configuration tools, common and advanced.

SSL in the Oracle Fusion Middleware Architecture

Review this topic for the role of SSL communication among the various components of the Oracle Fusion Middleware in its architecture.

Figure 6-2 SSL in Oracle Fusion Middleware



> **Note:**
>
> - In Figure 6-2, the label "Oracle Enterprise Manager" refers to the Fusion Middleware Control user interface.
> - Other administrative tools are available for specific tasks.

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

1. Enabling SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control and Enabling SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using WLST
2. Enabling SSL for Outbound Requests from Oracle HTTP Server
3. Configuring Inbound SSL to Oracle WebLogic Server
4. Outbound connections to the LDAP server can originate from Oracle Platform Security Services or from Oracle WebLogic Server:

   a. Configuring Outbound SSL from Oracle Platform Security Services to LDAP
   b. Configuring Outbound SSL from LDAP Authenticator to LDAP
5. SSL-Enabling a Data Source
6. SSL-Enabling Oracle Database
7. Client-Side SSL for Applications
8. Configuring Outbound SSL from Oracle WebLogic Server
9. SSL-Enabling Oracle Database

Keystores and Wallets

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

For details, see Keystores and Oracle Wallets.

Keystores and Oracle Wallets

Oracle Fusion Middleware 14*c* (14.1.2.0.0) supports different types of keystores for keys and certificates.

- JKS-based keystore and truststore

  Oracle WebLogic Server uses JKS keystores in upgraded environments.

  JDK's `keytool` utility manages JKS keystores and certificates.

- Oracle wallet

  System components like Oracle HTTP Server use the Oracle wallet.

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

- OPSS Keystore Service (KSS) keystore and truststore

  The Keystore Service provides an alternative mechanism to manage keys and certificates. Oracle WebLogic Server uses KSS keystores out-of-the-box.

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

For more information about these types of stores, and when to use which type of store, see Keystores and Wallets.

> **See Also:**
>
> Key and Certificate Storage in Oracle Fusion Middleware for keystore management

JDK17 Requires keyUsage with keyCertSign

Under JDK17, self-signed CA certificates used for SSL configuration must have the keyUsage extension with keyCertSign asserted.

TLS Protocol Support in Oracle Fusion Middleware

Oracle Fusion Middleware supports TLS v1.2 cryptographic libraries. However, you will need to configure this protocol in the component since the library may not be enabled by default.

Table provides links to the relevant procedures:

Table 6-1 TLS v1.2 Support in Oracle Fusion Middleware Components

Component/Feature	TLS Documentation
Oracle HTTP Server	"SSLProtocol" in *Administering Oracle HTTP Server*
Oracle WebLogic Server	"Specifying the SSL Protocol Version" in *Administering Security for Oracle WebLogic Server*

Authentication Modes in Oracle Fusion Middleware

Review this topic for the different modes of authentication that are supported between a client and a server in Oracle Fusion Middleware.

The following authentication modes are supported:

- In *no-authentication mode*, neither server nor client are required to authenticate.

  Other names for this mode include Anonymous SSL/No Authentication/Diffie-Hellman. This mode is considered unsecured.
- In *server authentication mode*, a server authenticates itself to a client.

  This mode is also referred to as One-way SSL/Server Authentication.
- In *mutual authentication mode*, a client authenticates itself to a server and that server authenticates itself to the client.

  This mode is also known as Two-way SSL/Client Authentication.
- In *optional client authentication mode*, the server authenticates itself to the client, but the client may or may not authenticate itself to the server. Even if the client does not authenticate itself, the SSL session still goes through.

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 Remote Console
- `keytool` command-line tool

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

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

Advanced Tools

`orapki` command-line tool is needed to manage wallets for certain stand-alone installations.

> **See Also:**
>
> Key and Certificate Storage in Oracle Fusion Middleware for keystore management.

Configuring SSL for Configuration Tools

Several tools are available for Oracle Fusion Middleware configuration. These tools must be configured with SSL to enable them to connect to an SSL-enabled Oracle WebLogic Server.

> **See Also:**
>
> Configuring Inbound SSL to Oracle WebLogic Server for details about enabling inbound SSL on Oracle WebLogic Server.

For a list of all the configuration tools, see Tools for SSL Configuration.

This section contains these topics:

- Oracle Enterprise Manager Fusion Middleware Control
  Follow these steps to launch the Oracle Fusion Middleware Control or Enterprise Manager.
- Oracle WebLogic Remote Console
  Follow these steps to launch the Oracle WebLogic Remote Console.
- WLST Command-Line Tool
  Follow these steps to launch WLST for configuring SSL.
- orapki Utility
  orapki is the recommended tool to configure wallets.

Oracle Enterprise Manager Fusion Middleware Control

Follow these steps to launch the Oracle Fusion Middleware Control or Enterprise Manager.

1. 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.

   For details, see “Configure server SSL settings in *Administering Oracle WebLogic Server with Fusion Middleware Control*.

2. Now launch Fusion Middleware Control using an SSL-based URL, in the format https://*host:port*.

Oracle WebLogic Remote Console

Follow these steps to launch the Oracle WebLogic Remote Console.

Ensure that the SSL port is enabled on the Oracle WebLogic Server instance. Now launch the Remote Console. You may get a warning that the certificate is not trusted; accept this certificate and continue.

For details, see “Configure server SSL settings” in *Administering Oracle WebLogic Server with Fusion Middleware Control*.

For more information, see Oracle WebLogic Remote Console Online Help.

WLST Command-Line Tool

Follow these steps to launch WLST for configuring SSL.

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

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

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

> **See Also:**
>
> WLST Reference for SSL for details about using WLST.

orapki Utility

orapki is the recommended tool to configure wallets.

For details, see orapki.

Configuring SSL for the Web Tier

Oracle HTTP Server resides in the web tier of the Oracle Fusion Middleware. It secures communications by using a SSL protocol.

This section describes SSL for Oracle HTTP Server which resides in the Web tier, and contains these topics:

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

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

This chapter does not cover all Oracle HTTP Server configuration options. For additional scenarios, see “OHS in a WebLogic Server Domain” and “OHS in a Standalone Domain” in *Installing and Configuring Oracle HTTP Server*.

- Configuring Load Balancers
  Use the instructions specific to your load-balancing device to configure load balancers in your Oracle Fusion Middleware environment.
- Enabling SSL for Oracle HTTP Server Virtual Hosts
  Find out how to manage SSL configuration for Oracle HTTP Server virtual hosts operating in an Oracle WebLogic Server environment.

Configuring Load Balancers

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

Enabling SSL for Oracle HTTP Server Virtual Hosts

Find out how to manage SSL configuration for Oracle HTTP Server virtual hosts operating in an Oracle WebLogic Server environment.

> **Note:**
>
> For Oracle HTTP Server in standalone mode, see “Configuring Secure Sockets Layer in Standalone Mode” in *Administering Oracle HTTP Server*.

For outbound traffic, see Enabling SSL for Outbound Requests from Oracle HTTP Server (using either Fusion Middleware Control or WLST).

- Enabling SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using Fusion Middleware Control
- Enabling SSL for Inbound Requests to Oracle HTTP Server Virtual Hosts Using WLST
- Enabling SSL for Outbound Requests from Oracle HTTP Server

Enabling 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 Managing Keystores, Wallets, and Certificates.

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

   The **Virtual Hosts** page shows what hosts are currently configured, and whether they are configured for http or https.

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

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

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

   - Check the **Enable SSL** box.
   - Select a wallet from the drop-down list.
   - From the Server SSL properties, select the SSL authentication type, cipher suites to use, and the SSL protocol version.

   > **Note:**
   >
   > The default values are appropriate in most situations.

   > **Note:**
   >
   > – This assumes that the certificate is available in Fusion Middleware Control. If it was created through orapki, import it first as explained in “Importing a Keystore” in *Securing Applications with Oracle Platform Security Services*.
   >
   > – 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 "Properties Files for SSL" in *WLST Command Reference for Infrastructure Security*.

6. Click **OK** to apply the changes.

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

8. Restart the Oracle HTTP Server instance by navigating to **Oracle HTTP Server**, then **Control**, then **Restart.**

9. Open a browser session and connect to the port number that was SSL-enabled.

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

Execute these WLST commands using the protocol described in How to Launch the Command-Line Interface.

Take these steps:

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

   ```
   listListeners('OHS_instance','OHS_instance' )
   ```

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

2. Configure the virtual host with SSL properties:

   ```
   configureSSL('OHS_instance',
      'OHS_instance',
      'ohs',
      'vhost1')
   ```

   > **Note:**
   >
   > - `configureSSL` uses defaults for all SSL attributes; see "Default Values of Parameters" in *WLST Command Reference for Infrastructure Security* for details.
   > - You could also specify a properties file as a parameter to `configureSSL`. See "Parameters in Properties File" and `"configureSSL"` in the *WLST Command Reference for Infrastructure Security* for details and examples of how to use a properties file. See `"configureSSL"` in the same document for details about this command.

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

Enabling SSL for Outbound Requests from Oracle HTTP Server

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

- Enabling One-Way SSL
- Enabling Two-Way SSL

Enabling One-Way SSL

The steps are as follows:

1. Generate a custom keystore for Oracle WebLogic Server (see Configuring SSL for Oracle WebLogic Server) containing a certificate.

2. Import the trusted CA certificate used by Oracle WebLogic Server into the Oracle HTTP Server wallet as a trusted certificate. You can use any available utility such as WLST or Fusion Middleware Control for this task. (*Note*: The wallet mentioned here is the one that the Oracle HTTP Server listen port uses for SSL. The trusted (root) CA certificate that signed the Oracle WebLogic Server certificate must exist in this wallet. For details on importing trusted certificates see “Importing a Certificate Using Fusion Middleware Control” in *Securing Applications with Oracle Platform Security Services*.)
3. With Oracle WebLogic Server instance shut down, edit the Oracle HTTP Server configuration file *DOMAIN_HOME*/config/fmwconfig/components/OHS/*instance_name*/ssl.conf and add the following line to the SSL configuration under `mod_weblogic`:

   ```
   WlSSLWallet  "$(DOMAIN_HOME}/config/fmwconfig/components/COMPONENT_TYPE/
   COMPONENT_NAME/keystores/default"
   ```

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

   Here is an example of how the configuration should look:

   ```
   <IfModule mod_weblogic.c>
   WebLogicHost myweblogic.server.com
   WebLogicPort 7002
   MatchExpression *.jsp
   SecureProxy On
   WlSSLWallet "$(DOMAIN_HOME}/config/fmwconfig/components/OHS/ohs1/keystores/default"
   </IfModule>
   ```

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

Enabling Two-Way SSL

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

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

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

   You can use any available utility such as WLST or Fusion Middleware Control for export, and the `keytool` utility for import. For details, see Managing the Certificate Life Cycle.
4. From the Oracle WebLogic Remote Console, select the **Keystores** tab for the server being configured.
5. Set the custom trust store with the `trust.jks` file location of the trust store that was created in Step 2 (use the full name).

6. Set the keystore type as JKS, and set the passphrase used to create the keystore.
7. Under the **SSL** tab, ensure that Trusted Certificate Authorities is set as **from Custom Trust Keystore**.
8. Ensure that Oracle WebLogic Server is configured for two-way SSL. For details, see "Configuring SSL" in *Administering Security for Oracle WebLogic Server*.

Configuring SSL for the Middle Tier

SSL in the middle tier of the Oracle Fusion Middleware includes enabling SSL for the application server and also the components and applications running on the application server.

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
  Follow these procedures to configure the application server.
- Client-Side SSL for Applications
  Find out how to enable SSL for applications on the client side.

Configuring SSL for Oracle WebLogic Server

Follow these procedures to configure the application server.

- Configuring Inbound SSL to Oracle WebLogic Server
- Configuring Outbound SSL from Oracle WebLogic Server

Configuring Inbound SSL to Oracle WebLogic Server

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

- "Configuring Oracle OPSS Keystore Service "
- "Overview of Configuring SSL in WebLogic Server"

Configuring Outbound SSL from Oracle WebLogic Server

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

- Configuring Outbound SSL from Oracle Platform Security Services to LDAP
- Configuring Outbound SSL from Oracle Platform Security Services to Oracle Database
- Configuring Outbound SSL from LDAP Authenticator to LDAP
- Configuring Outbound SSL to the Database

Configuring Outbound SSL from Oracle Platform Security Services to LDAP

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

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

Anonymous SSL (Server-side)

Start the LDAP server in anonymous authentication mode.

Consult your LDAP server documentation for information on this task.

Anonymous SSL (Client-side)

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

One-Way SSL (Client-side)

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

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

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

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

Configuring Outbound SSL from Oracle Platform Security Services to Oracle Database

You can set up a one-way or two-way SSL connection to a database-based OPSS security store.

For details about configuring the database server and clients, see “Setting Up an SSL Connection to the Database Security Store” in *Securing Applications with Oracle Platform Security Services*.

Configuring Outbound SSL from LDAP Authenticator to LDAP

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

Configuring Outbound SSL to the Database

For more information about configuring SSL for the database listener, see *Oracle Database Advanced Security Administrator's Guide*.

Client-Side SSL for Applications

Find out how to enable SSL for applications on the client side.

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

For best practices, refer to Best Practices for Application Developers.

Configuring SSL for the Data Tier

The data tier of the Oracle Fusion Middleware includes Oracle Database as a component. All components in the data tier must be SSL enabled.

- Configuring SSL for the Database
  Follow these procedures to enable SSL in the Oracle Database and the Data Sources on Oracle WebLogic Server.

Configuring SSL for the Database

Follow these procedures to enable SSL in the Oracle Database and the Data Sources on Oracle WebLogic Server.

- SSL-Enabling Oracle Database
- SSL-Enabling a Data Source

SSL-Enabling 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 Changing a Self-Signed Wallet to a Third-Party Wallet.

```
mkdir root
mkdir server

# Create root wallet, add self-signed certificate and export
orapki wallet create -wallet ./root -pwd password
orapki wallet add -wallet ./root -dn CN=root_test,C=US -keysize 2048 -self_signed -
validity 3650 -pwd password
orapki wallet display -wallet ./root -pwd password
orapki wallet export -wallet ./root -dn CN=root_test,C=US -cert ./root/
b64certificate.txt -pwd password

#Create server wallet, add self-signed certificate and export
orapki wallet create -wallet ./server -pwd password
orapki wallet add -wallet ./server -dn CN=server_test,C=US -keysize 2048 -pwd
password
orapki wallet display -wallet ./server -pwd password
orapki wallet export -wallet ./server -dn CN=server_test,C=US -request ./server/
creq.txt -pwd password

# Import trusted certificates
orapki cert create -wallet ./root -request ./server/creq.txt -cert ./server/cert.txt
-validity 3650 -pwd password
orapki cert display -cert ./server/cert.txt -complete
orapki wallet add -wallet ./server -trusted_cert -cert ./root/b64certificate.txt -
pwd password
```

```
orapki wallet add -wallet ./server -user_cert -cert ./server/cert.txt -pwd password
orapki wallet create -wallet ./server -auto_login -pwd password}}
```

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

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

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

WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/wallet_location)))

SSL_CLIENT_AUTHENTICATION=FALSE}}
```

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

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = dbname)
      (ORACLE_HOME = /path_to_O_H)
      (SID_NAME = sid)
    )
  )

SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /wallet_path)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = mynode.mycorp.com)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCPS)(HOST = mycorp.com)(PORT = 2490))
    )
  )
```

   Note that the SSL port has been added.

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

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

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

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

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

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

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

SSL-Enabling 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 WebLogic Remote Console, navigate to the data source that you are using.

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

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

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

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

- If you will require no client authentication:

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

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

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

   ```
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS)(HOST=host-name)
   (PORT=port-number)))(CONNECT_DATA=(SERVICE_NAME=service))
   (SECURITY=(SSL_SERVER_CERT_DN="CN=server_test,C=US")))
   ```

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

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

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

Setting Up One-Way SSL to the LDAP Security Store

Follow these steps to set up a one-way Secure Sockets Layer (SSL) channel between Oracle WebLogic Server or a Java SE application, and the LDAP security store. Such connection may be required, for example, when reassociating to an LDAP target store.

Configure LDAP

For information about how to configure LDAP with one-way SSL, see Enabling SSL on Oracle Internet Directory Listeners in *Administrator's Guide for Oracle Internet Directory*.

Create the LDAP Certificate Authority

If the LDAP certificate authority is unknown to WebLogic Server, then use `orapki` to export a certificate:

```
orapki wallet export -wallet CA -dn "CN=myCA" -cert serverTrust.cert
```

This command exports a certificate from a wallet to a file that is specified by `–cert`.

Before configuring SSL, note that:

- The following procedures are required if the type of SLL is `server-auth` and `mutual-auth` and not required for `no-auth`.
- If the flags specified in the procedures below are used in an environment where multiple applications run, then these applications must use the same truststore.

Setting Up for Jakarta EE Applications

Use one of the following procedures to set up a one-way SSL connection between the server and the identity store. The procedures differ because the identity store and security store services use different socket factories.

To establish one-way SSL connections between the server and the identity store (if applicable, then it is assumed that the trust certification authority (CA) has been exported):

1. If the CA is known to WebLogic Server, then skip this step.

   Otherwise, use `keytool` to import the LDAP CA into the truststore as in the following example which generates the `myKeys.jks` file and imports the `serverTrust.cert` file:

```
>keytool -import -v -trustcacerts -alias trust -file serverTrust.cert -keystore
myKeys.jks -storepass keyStorePassword
```

2. Modify the `startWebLogic.sh` script that starts the server to include a line like the following, and then run the script:

```
-Djavax.net.ssl.trustStore=<absolute path name to file myKeys.jks>
```

To establish a one-way SSL connection between the server and the security store (if applicable, then it is assumed that the trust CA has been exported):

1. Use `keytool` to import trust CA to the truststore:

```
>keytool -import -v -trustcacerts -alias trust -file serverTrust.cert -keystore
myKeys.jks -storepass keyStorePassword
```

2. Modify the `startWebLogic.sh` script that starts the server to include a line like the following, and then run the script:

```
-Dweblogic.security.SSL.trustedCAKeyStore=<absolute path name to file myKeys.jks>
```

3. If the LDAP server uses a wild card in the SSL certificate, then add the following line to the script that starts WebLogic Server:

```
-Dweblogic.security.SSL.ignoreHostnameVerification=true
```

4. Restart the server.

Setting Up for Java SE Applications

1. If the CA is known to WebLogic Server, then skip this step.

   Otherwise, use `keytool` to import the LDAP CA into the truststore as in the following example, which generates the `myKeys.jks` file and imports the `serverTrust.cert` file:

```
>keytool -import -v -trustcacerts -alias trust -file serverTrust.cert -keystore
myKeys.jks -storepass keyStorePasswodr
```

2. Modify the script that starts the Java Virtual Machine (JVM) to include a line like the following:

```
-Djavax.net.ssl.trustStore=<absolute path name to file myKeys.jks>
```

3. Restart the server.

Setting Up SSL in Identity Store Services

Establishing SSL connections in Identity Store Services using libOVD and JKS requires keystore certificates that are kept in multiple locations, such as the WebLogic Server truststore and the `adapters.jks` file.

Table 6-2 SSL with JKS

Virtualize Flag	Using the User and Role API	Using the Identity Directory API
`virtualize=false`	Specify the truststore as explained in Setting Up One-Way SSL to the LDAP Security Store.	Use `adapters.jks` as shown in Setting up One-Way SSL in Identity Store Services using libOVD and JKS and Setting up Two-Way SSL in Identity Store Services using libOVD and JKS.

Table 6-2 (Cont.) SSL with JKS

Virtualize Flag	Using the User and Role API	Using the Identity Directory API
`virtualize=true`	Use `adapters.jks` as shown in Setting up One-Way SSL in Identity Store Services using libOVD and JKS and Setting up Two-Way SSL in Identity Store Services using libOVD and JKS.	Use `adapters.jks` as shown in Setting up One-Way SSL in Identity Store Services using libOVD and JKS and Setting up Two-Way SSL in Identity Store Services using libOVD and JKS.

- Setting up One-Way SSL in Identity Store Services using libOVD and JKS
  Follow these steps to establish one-way SSL in Identity Store Services using libOVD and JKS.
- Setting up Two-Way SSL in Identity Store Services using libOVD and JKS
  Follow these steps to establish two-way SSL in Identity Store Services using libOVD and JKS.

Setting up One-Way SSL in Identity Store Services using libOVD and JKS

Follow these steps to establish one-way SSL in Identity Store Services using libOVD and JKS.

1. Create a keystore to contain the LDAP server certificate(s) for use by the service. You must provide passwords for Oracle WebLogic Remote Console and the keystore, respectively. Before running `libovdconfig.sh`, set `ORACLE_HOME` to `oracle_common`.

   To create the keystore run `MW_HOME/oracle_common/bin/libovdconfig.sh` with the `createKeystore` option:

   ```
   libovdconfig.sh -host wls_host -port wls_adminserver_port -userName wls_user_name -
   domainPath full_path_domain_home -createKeystore
   ```

   where:

   - `host` is the server host
   - `port` is the WebLogic Administration Server port
   - `username` is the administrator user name
   - `domainPath` is the complete path to the domain home
2. Export the certificate from the LDAP directory with the LDAP utility export.
3. Import the certificate to the created keystore with the keytool command:

   ```
   $JAVA_HOME/bin/keytool -importcert
   -keystore $DOMAIN_HOME/config/fmwconfig/ovd/default/keystores/adapters.jks
   -storepass keystore_password_used_in_libovdconfig.sh
   -alias alias_name
   -file full_path_to_LDAPCert_file
   -noprompt
   ```
4. Restart WebLogic Server.

Setting up Two-Way SSL in Identity Store Services using libOVD and JKS

Follow these steps to establish two-way SSL in Identity Store Services using libOVD and JKS.

1. Perform the procedure described in Setting up One-Way SSL in Identity Store Services using libOVD and JKS.
2. In the keystore that was created, generate a new signed certificate.
3. Export this certificate to a file.
4. Import the certificate into the LDAP directory.

Advanced SSL Scenarios

SSL configuration in Oracle Fusion Middleware also includes additional scenarios that are beyond the basic topologies like FIPS 140 support, Hardware Security Modules and certificate validation.

- 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.
- CRL Integration with SSL
  Find out how to configure a component to use CRL-based validation, and how to create and set up CRLs on the file system.
- Oracle Fusion Middleware FIPS 140-2 Settings
  Review this topic for setting up FIPS 140–2 among the different components of the Oracle Fusion Middleware.

Hardware Security Modules and Accelerators

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

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

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

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

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

      On Linux, the library is located at:

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

On Windows, the library is located at:

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

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

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

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

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

> **See Also:**
>
> 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.

CRL Integration with SSL

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

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

Components that use SSL can optionally turn on certificate validation using a certificate revocation list (CRL). This allows them to validate the peer certificate in the SSL handshake

and ensure that it is not on the list of revoked certificates issued by the Certificate Authority (CA).

- Configuring CRL Validation for a Component
- Manage CRLs on the File System
- Test a Component Configured for CRL Validation

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 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
```

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 Managing Certificate Revocation Lists with orapki Utility.

CRL Renaming to Hashed Form

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

To rename a CRL:

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

For example:

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

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

CRL Creation

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

To create a CRL:

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

For example:

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

Certificate Revocation

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

To revoke a certificate:

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

For example:

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

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 Manage CRLs on the File System.
3. Configure your component to use this CRL. Follow the steps outlined in Configuring CRL Validation for a Component.
4. The SSL handshake should fail when this revoked certificate is used.

Oracle Fusion Middleware FIPS 140-2 Settings

Review this topic for setting up FIPS 140–2 among the different components of the Oracle Fusion Middleware.

For details about FIPS 140 support in Oracle Fusion Middleware, see FIPS 140 Support in Oracle Fusion Middleware.

Best Practices for SSL

Oracle Fusion Middleware recommends some best practices for component administrators and application developers while configuring SSL.

- Best Practices for Administrators
  For a successful system administration, you must follow these best practices.
- Best Practices for Application Developers
  Application Developers must follow these practices that are recommended.

Best Practices for Administrators

For a successful system administration, you must follow these best practices.

- 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 Managing Keystores, Wallets, and Certificates.
- It is recommended that components (at least in the Web tier) use certificates that have the system host name or virtual host or site name as the DN. This allows browsers to connect in SSL mode without giving unsettling warning messages.
- A minimum key size of 1024 bits is recommended for certificates used for SSL. Higher key size provides more security but at the cost of reduced performance. Pick an appropriate key size value depending on your security and performance requirements.
- Lack of trust is one of the most common reasons for SSL handshake failures. Ensure that the client trusts the server (by importing the server CA certificate into the client keystore) before starting SSL handshake. If client authentication is also required, then the reverse should also be true.

Best Practices for Application Developers

Application Developers must follow these practices that are recommended.

- Use Java Key Store (JKS) to store certificates for your Jakarta 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.

WLST Reference for SSL

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

> **See Also:**
>
> 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.

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

7

Managing Keystores, Wallets, and Certificates

Oracle Fusion Middleware supports security features and tools to administer keystores, keys, and certificates. These artifacts are used to configure SSL and related tasks for Oracle Fusion Middleware components.

- Key and Certificate Storage in Oracle Fusion Middleware
  Keys and certificates are used to digitally sign and verify data and achieve authentication, integrity, and privacy in network communications.
- Command-Line Interface for Keystores and Wallets
  Oracle Fusion Middleware provides a set of `WLST` scripts to create and manage keystores and Oracle wallets, and to manipulate their stored objects.
- Keystore Management
  The keystore service allows you to manage and administer keys and certificates for Secure Sockets Layer (SSL), message security, encryption, and other tasks that require special certificates.
- Managing Keystores with Fusion Middleware Control
- Wallet Management
  Oracle wallets provide a full range of management features to enable you to create, maintain, and delete certificates for various applications.

Key and Certificate Storage in Oracle Fusion Middleware

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

Private keys, digital certificates, and trusted CA certificates are stored in keystores. This section describes the keystores available in Oracle Fusion Middleware and contains these topics:

- Types of Keystores
  Oracle Fusion Middleware provides various types of keystores for keys and certificates.
- Keystore Management Tools
  Oracle Fusion Middleware provides WLST, orapki and Fusion Middleware Control as options for keystore operations.

Types of Keystores

Oracle Fusion Middleware provides various types of keystores for keys and certificates.

The various types of keystores as shown in Table 7-1:

Table 7-1 Keystore Types in Oracle Fusion Middleware

Keystore Type	Description	Protection Mechanism
Oracle Wallet	Oracle Wallet	Password or auto-login

Table 7-1 (Cont.) Keystore Types in Oracle Fusion Middleware

Keystore Type	Description	Protection Mechanism
PKCS12	Java Keystore	Password
JKS	Java Keystore	Password
KSS	OPSS Keystore Service	Password or policy

- About Oracle Wallet
  An Oracle wallet is a container that stores your credentials, such as certificates, trusted certificates, certificate requests, and private keys. You can store Oracle wallets on the file system or in LDAP directories such as Oracle Internet Directory. Oracle wallets can be auto-login or password-protected wallets.
- About the JKS Keystore
  A JKS keystore is the default JDK implementation of Java keystores. Jakarta EE applications can use the JKS-based keystore and truststore.
- About the Keystore Service (KSS) Keystore
  The OPSS Keystore Service enables you to manage keys and certificates for SSL, message security, encryption, and related tasks. The Keystore Service offers several advantages including policy-based protection and centralized management of keystores and truststores, expiring certificates, and other key material.

About Oracle Wallet

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

When creating a wallet, you can:

- Pre-populate it with a self-signed certificate. Such a wallet is called a test wallet and is typically used in development and testing phases.
- Create a certificate request, so that you can request a signed certificate back from a Certificate Authority (CA). Once the CA sends the certificate back, it is imported into the wallet. Such a wallet is called a third-party wallet.

Either the test wallet or the third-party wallet may be password-protected, or may be configured to not require a password, in which case it is called an auto-login wallet.

Oracle Wallets are used for Oracle HTTP Server. As of Oracle Fusion Middleware 14*c* (14.1.2.0.0), you can take advantage of the central storage and unified management available with the Keystore Service to manage wallets and their contents through the export, import, and synchronization features of that service.

> **See Also:**
>
> - About the Keystore Service (KSS) Keystore for more information on the Keystore Service.
> - "Infrastructure Security Custom WLST Commands" in *WLST Command Reference for Infrastructure Security* for details about the `importKeyStore`, `exportKeyStore`, and `syncKeyStore` commands.

About the JKS Keystore

A JKS keystore is the default JDK implementation of Java keystores. Jakarta EE applications can use the JKS-based keystore and truststore.

About the Keystore Service (KSS) Keystore

The OPSS Keystore Service enables you to manage keys and certificates for SSL, message security, encryption, and related tasks. The Keystore Service offers several advantages including policy-based protection and centralized management of keystores and truststores, expiring certificates, and other key material.

In Oracle Fusion Middleware 14*c* (14.1.2.0.0), Oracle WebLogic Server:

- uses the Keystore Service out-of-the-box
- uses JKS by default in upgraded environments.

Keystore Management Tools

Oracle Fusion Middleware provides WLST, orapki and Fusion Middleware Control as options for keystore operations.

About Importing DER-encoded Certificates

You cannot use Fusion Middleware Control or the `WLST` command-line tool to import DER-encoded certificates or trusted certificates into an Oracle wallet. Use `orapki` command-line tool instead.

Using a Keystore Not Created with WLST or Fusion Middleware Control

> **Note:**
>
> This is only applicable in a collocated environment.

If an Oracle wallet was created with tools such as `orapki`, it must be imported prior to use. Specifically for Oracle HTTP Server, if a wallet was created using `orapki`, in order to view or manage it in Fusion Middleware Control you must first import it with either Fusion Middleware Control or the WLST `importKeyStore` command. For details, see "Importing a Keystore" in *Securing Applications with Oracle Platform Security Services*.

Copying Keystores to File System Not Supported

> **Note:**
>
> This is only applicable in a collocated environment.

Creating, renaming, or copying keystores directly to any directory on the file system is not supported. Any pre-existing keystore or wallet that you wish to use must be imported using either Fusion Middleware Control or the WLST utility.

Managing Wallets in a Stand-alone Environment

In a standalone environment, such as a standalone OHS installation, orapki commands are used for wallet management. For details, see “Configuring SSL in a Standalone Mode” in *Administering Oracle HTTP Server*.

Additional Information

Details about the tools are provided in these sections:

- Command-Line Interface for Keystores and Wallets
- Wallet Management
- orapki

Command-Line Interface for Keystores and Wallets

Oracle Fusion Middleware provides a set of `WLST` scripts to create and manage keystores and Oracle wallets, and to manipulate their stored objects.

For more information on how to launch the command line interface, see How to Launch the Command-Line Interface.

- How to Launch the Command-Line Interface
  When running SSL `WLST` commands, you must invoke the `WLST` script from the Oracle Common home.

How to Launch the Command-Line Interface

When running SSL `WLST` commands, you must invoke the `WLST` script from the Oracle Common home.

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.

> **Note:**
>
> All SSL-related WLST commands require you to launch the script from the above-mentioned location only.

Here is the basic execution sequence:

```
./wlst.sh
connect('weblogic','<password>') --- To connect to WebLogic Admin Server
editCustom()
startEdit()
wlstCommand('param1', 'param2', 'param3', 'param4')
save()
activate()
```

where `wlstCommand` is the actual WLST command. For example, to create an OHS wallet:

```
connect('weblogic','<password>') --- For connecting to WLS Admin Server
editCustom()
startEdit()
createWallet('ohs1', 'ohs1', 'ohs', 'testwallet')
save()
activate()
```

In this command the first two parameters both refer to the component instance name (in contrast with earlier releases where they referred to an Oracle instance and a component instance respectively, in this release both refer to the latter), the third parameter is the component, and the fourth parameter is the wallet name.

Here is a sample output for `createWallet`:

```
wls:/base_domain/serverConfig> editCustom()
Location changed to edit custom tree. This is a writable tree with No root.
For more help, use help('editCustom')

wls:/base_domain/editCustom> startEdit()
Starting an edit session ...
Started edit session, please be sure to save and activate your
changes once you are done.
wls:/base_domain/editCustom> createWallet('ohs1','ohs1','ohs','testwallet','password')
Wallet created
wls:/base_domain/editCustom> save()
Saving all your changes ...
Saved all your changes successfully.
wls:/base_domain/editCustom> activate()
Activating all your changes, this may take a while ...
The edit lock associated with this edit session is released
once the activation is completed.
Activation completed
```

Keystore Management

The keystore service allows you to manage and administer keys and certificates for Secure Sockets Layer (SSL), message security, encryption, and other tasks that require special certificates.

Since 14c (14.1.2.0.0), Fusion Middleware recommends Keystore Service (KSS) for wallet and certificate management in a collocated scenario. For details about KSS keystore management, see "Managing Keys and Certificates" in *Securing Applications with Oracle Platform Security Services*.

Managing Keystores with Fusion Middleware Control

You can manage keystores by performing various tasks with Oracle Enterprise Manager Fusion Middleware Control. For more information, see Managing Keystores with Fusion Middleware Control.

Wallet Management

Oracle wallets provide a full range of management features to enable you to create, maintain, and delete certificates for various applications.

> **Note:**
>
> Since 12c (12.2.1), the "Wallet" option is no longer available in the Fusion Middleware Control Security menu. Fusion Middleware recommends the Keystore Service (KSS) instead for wallet management in a collocated scenario. However, in a standalone environment, you can manage a wallet only by using the orapki utility. There is no KSS support in standalone scenario.
>
> For wallet configuration in a stand-alone context, for example a stand-alone Oracle HTTP Server, see orapki.

This discussion assumes that components are installed within a WebLogic domain.

- About Wallets and Certificates
  Review these topics for various types of wallets and their recommendation, conventions and requirements.
- Managing the Wallet Life Cycle
  Review this topic for the typical life cycle events for an Oracle wallet.
- Common Wallet Operations
  Follow these procedures for the steps required to perform a range of wallet management functions.
- Managing the Certificate Life Cycle
  Review this topic for the typical life cycle event of a certificate, starting from wallet creation.
- Common Certificate Operations
  Follow these procedures for the steps required to perform a range of certificate management functions.
- Wallet and Certificate Maintenance
  Review these topics for more information on maintaining wallet and certificates.
- Managing Certificates with Fusion Middleware Control

About Wallets and Certificates

Review these topics for various types of wallets and their recommendation, conventions and requirements.

This section contains the following topics:

- About Password-Protected and Autologin Wallets

- About 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.
- Sharing Wallets Across Instances
  Oracle recommends that you do not share wallets between component instances, since each wallet represents a unique identity.
- Wallet Naming Conventions

About 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 14*c* (14.1.2.0.0), 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 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. The wallet must be modified 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.

About 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 Certificate Authorities (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.

Sharing Wallets Across Instances

Oracle recommends that you do not share wallets between component instances, since each wallet represents a unique identity.

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

Note that no management tools or interfaces are available to facilitate wallet sharing. However, you can export a wallet from one instance and import it into another instance. See Common Wallet Operations for details of wallet export and import.

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, 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.

Managing the Wallet Life Cycle

Review this topic for the typical life cycle events for an Oracle wallet.

- 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.

> **Note:**
>
> As of Oracle Fusion Middleware 14c (14.1.2.0.0), you can take advantage of the central storage and unified console available with the Keystore Service to manage wallets and their contents through the export, import, and synchronization features of that service. See “Infrastructure Security Custom WLST Commands” in *WLST Command Reference for Infrastructure Security* for details about the `importKeyStore`, `exportKeyStore`, and `syncKeyStore` commands.

Common Wallet Operations

Follow these procedures for the steps required to perform a range of wallet management functions.

- Creating a Wallet Using orapki
- Adding a Self-Signed Certificate to a Wallet Using orapki

Creating a Wallet Using orapki

To create a Password Protected Wallet (ewallet.p12 and cwallet.sso):

```
orapki wallet create -wallet wallet_location -auto_login
```

To create an *Auto-Login Wallet* (cwallet.sso only):

```
orapki wallet create -wallet wallet_location -auto_login_only
```

For more information on using the orapki utility for creating a wallet, see Creating and Viewing Oracle Wallets with orapki.

Adding a Self-Signed Certificate to a Wallet Using orapki

To create a wallet containing a self-signed certificate using orapki:

```
orapki wallet add -wallet wallet_location -dn user_dn -keysize 512|1024|2048|4096|8192|
16384 -self_signed [-pwd][-auto_login_only]
```

> **Note:**
>
> If alias value is not specified for a wallet, a default alias will be set. Example: `orakey,orakey1,orakey2.`

To create a wallet containing self-signed certificate with ECC keys:

```
orapki wallet add -wallet wallet_location -dn user_dn -sign_alg signing_alg -asym_alg
ECC -eccurve curve_type
```

For more information on adding an ECC certificate request to an Oracle Wallet, see Adding an ECC Certificate Request to an Oracle Wallet.

Managing the Certificate Life Cycle

Review this topic for the typical life cycle event of a certificate, starting from wallet creation.

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 “Replacing Demonstration CA Signed Certificates” in *Securing Applications with Oracle Platform Security Services*.

Common Certificate Operations

Follow these procedures for the steps required to perform a range of certificate management functions.

- Adding a Certificate Request Using orapki
- Exporting a Certificate from an Oracle Wallet using orapki
- Exporting a Certificate Request Using orapki
- Importing a Trusted Certificate Using orapki
- Importing a User Certificate Using orapki
- Creating a Signed Certificate from Certificate Requests Using orapki

Adding a Certificate Request Using orapki

To add a certificate signing request to an Oracle Wallet:

```
orapki wallet add -wallet wallet_location -dn user_dn -keysize certificate_key_size -
addext_ski -addext_ku extension_key_usage -addext_basic_cons CA -pathLen number -
addext_san DNS [-pwd] [-auto_login_only]
```

> **Note:**
>
> In all commands where `user_dn` is referenced, use single quotes for UNIX and double quotes for Windows.

For more information on using the orapki utility for adding certificate request to an Oracle wallet, see Adding Certificates and Certificate Requests to Oracle Wallets with orapki.

Exporting a Certificate from an Oracle Wallet using orapki

To export a certificate from an Oracle Wallet:

```
orapki wallet export -wallet wallet_location -dn
certificate_dn -cert certificate_filename -issuer_dn dn_of_issuer -serial_num
serial_number_of_certificate
```

For more information on using the orapki utility for exporting a certificate from an Oracle Wallet, see Exporting Certificates and Certificate Requests from Oracle Wallets with orapki.

Exporting a Certificate Request Using orapki

To export a certificate request from an Oracle wallet:

```
orapki wallet export -wallet wallet_location -dn certificate_request_dn -request
certificate_request_filename [-pwd]
```

For more information on using the orapki utility for exporting certificate requests from an Oracle Wallet, see Exporting Certificates and Certificate Requests from Oracle Wallets with orapki.

> **Note:**
>
> For detailed information about Trusted Certificates and how to identify the correct Trusted Root Certificate Authority Certificate(s) for a User Certificate, refer to support Document 1368940.1 on My Oracle Support. You can access My Oracle Support at: https://support.oracle.com/.

Importing a Trusted Certificate Using orapki

To import a trusted certificate into the Wallet:

```
orapki wallet add -wallet wallet_location -trusted_cert -cert certificate_location [-
pwd] [-auto_login_only]
```

For more information on using the orapki utility to import a trusted certificate to an Oracle wallet, see Adding Certificates and Certificate Requests to Oracle Wallets with orapki.

Importing a User Certificate Using orapki

To add a user certificate to an Oracle wallet:

```
orapki wallet add -wallet wallet_location -user_cert -cert certificate_location [-pwd]
[auto_login_only]
```

For more information on using the orapki utility to import a trusted certificate to an Oracle wallet, see Adding Certificates and Certificate Requests to Oracle Wallets with orapki.

Creating a Signed Certificate from Certificate Requests Using orapki

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]
```

For more information on using the orapki utility for creating a signed certificate, see Creating Signed Certificates for Testing Purposes.

Wallet and Certificate Maintenance

Review these topics for more information on maintaining wallet and certificates.

- Location of Wallets
- Effect of Host Name Change on a Wallet
- 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).
- Replacing an Expiring Certificate in a Wallet
  An expiring certificate in a wallet is replaced before it actually expires to avoid or reduce application downtime.

Location of Wallets

This section describes the location of wallets and provides maintenance details.

Root Directory for an Oracle HTTP Server Wallet

The root directory for Oracle HTTP Server wallets is:

`$DOMAIN_HOME/config/fmwconfig/components/OHS/`*`ohs_instance_name`*`/keystores`

This root directory contains subdirectories with wallet names, and these subdirectories contain the actual wallet files.

For example, assuming `ohs_instance1` contains two wallets named `ohs1` and `ohs2`, respectively. `ohs1` is a password-protected wallet, and `ohs2` is an auto-login-only wallet. A sample structure could look like this:

```
$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance1
/keystores/ohs1/cwallet.sso

$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance1
/keystores/ohs1/ewallet.p12

$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance1
/keystores/ohs2/cwallet.sso
```

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).

- Requesting a New Certificate for a Production Wallet
- Requesting a New Certificate for a Self-signed Wallet

Requesting a New Certificate for a Production Wallet

You can request a new certificate to import into a production wallet.

The steps are:

1. Generate a new request with the new DN (based on new host name).
2. Send this request to a certificate authority (CA).
3. Get back a new certificate from the CA.
4. Delete the older certificate and certificate request from the wallet.
5. Import the new certificate.

See Common Wallet Operations for details about these operations.

Requesting a New Certificate for a Self-signed Wallet

To add a new certificate to a self-signed wallet, you need to create a new wallet for the certificate.

The steps are:

1. Delete the existing wallet.
2. Create a new wallet with a self-signed certificate using the new DN (based on the new host name).

See Common Wallet Operations for details about these operations.

For both production and self-signed wallets, once the new certificate is available in the wallet, you need to ensure that it is imported into all the component wallets where it needs to be trusted. For example, if Oracle WebLogic Server is SSL-enabled and the certificate for Oracle WebLogic Server changed due to a host name change, then you need to import its new certificate into the Oracle HTTP Server wallet so that it can trust its new peer.

Changing a Self-Signed Wallet to a Third-Party Wallet

You can convert a self-signed wallet into a third-party wallet, one that contains certificates signed by a trusted Certificate Authority (CA).

Assuming a self-signed wallet named `MYWallet`, containing a certificate with DN as "`CN=my.example.com,O=example`", take these steps to convert it into a third-party wallet:

1. Remove the user certificate "`CN=my.example.com,O=example`" from the wallet.
2. Remove the trusted certificate "`CN=my.example.com,O=example`" from the wallet (this has the same DN as the user certificate, but is a separate entity nonetheless).

3. Export the certificate request "`CN=my.example.com,O=example`" from the wallet and save it to a file.
4. Give this certificate request file to a third-party certificate authority (CA) such as Verisign.
5. The CA will return one of the following:
   - A user certificate file and its own certificate file
   - A single file with a certificate chain consisting of a user certificate and its own certificate
6. Import the above file(s) into the wallet.

See Common Wallet Operations for details about these operations.

Replacing an Expiring Certificate in a Wallet

An expiring certificate in a wallet is replaced before it actually expires to avoid or reduce application downtime.

The steps for replacing an expiring certificate are as follows:

1. Export the certificate request from the wallet (this is the same request for which the current expiring certificate was issued).
2. Provide this certificate request to the third-party Certificate Authority (CA) for certificate issuance. The validity date of the new certificate should be earlier than the expiration date of the current certificate. This overlap is recommended to reduce downtime.

> **Note:**
>
> Steps 1 and 2 are not required when the third-party CA already maintains the certificate request in a repository. In that case, simply request the CA to issue a new certificate for that certificate request.

3. Remove the existing certificate (the one that is about to expire) from the wallet.
4. Import the newly issued certificate into the wallet.

   To reduce downtime, remove the previous certificate and import the new certificate in the overlap period when the new certificate has become valid and the older one has not yet expired.
5. If the new certificate was issued by a CA other than the one that issued the original certificate, you may also need to import the new CA's trusted certificate before importing the newly issued certificate.

See Common Wallet Operations for details about these operations.

Managing Certificates with Fusion Middleware Control

You can manage certificates by performing various tasks with Oracle Enterprise Manager Fusion Middleware Control. For more information, see Managing Certificates with Fusion Middleware Control.

8

FIPS 140 Support in Oracle Fusion Middleware

Oracle Fusion Middleware supports Federal Information Processing Standard, FIPS 140–2, a U.S. government standard that defines security requirements for cryptographic modules.

- About the FIPS Standard
  Federal Information Processing Standards (FIPS) are a series of standards established by the US National Institute of Standards for Technology (NIST) for use in evaluating the security of computer systems and networks.
- About FIPS 140-2 in Oracle Fusion Middleware Release 14c
  Oracle Fusion Middleware Release 14*c* supports the use of FIPS 140-2-enabled cryptographic libraries.
- Components with FIPS 140 Support
  When you plan to work with FIPS 140 in Oracle Fusion Middleware, be aware of the different components at various layers of the middleware stack where certain features may operate in FIPS 140 mode. If any component in the stack is operating in non-FIPS 140 mode, the transaction may not be FIPS 140-compliant. It is therefore important to ensure that all relevant components are operating in FIPS 140 mode.
- Common Scenarios for an Operational FIPS 140-2 Environment
  The implementation of a feature and the establishment of a connection between a client and a server are some of the possible scenarios you may use or encounter while operating in a FIPS 140–2 environment. Each component scenario uses a corresponding communication protocol and signature algorithm.
- Troubleshooting FIPS 140 Issues
  You may encounter problems while configuring FIPS 140 for different components of the Oracle Fusion Middleware and need information on how to troubleshoot those problems.

About the FIPS Standard

Federal Information Processing Standards (FIPS) are a series of standards established by the US National Institute of Standards for Technology (NIST) for use in evaluating the security of computer systems and networks.

One of the FIPS standards, FIPS 140-2, specifies the security requirements that must be met by a cryptographic module to protect sensitive information. The standard provides four increasing, qualitative levels of security to cover the wide range of potential applications and environments in which cryptographic modules may be employed.

> **Note:**
>
> In the remainder of this chapter, the term 'FIPS 140' refers to the FIPS 140-2 standard.

About FIPS 140-2 in Oracle Fusion Middleware Release 14c

Oracle Fusion Middleware Release 14*c* supports the use of FIPS 140-2-enabled cryptographic libraries.

The ability to operate in FIPS 140 mode is **not** a generic, product suite-wide claim. Instead, it is specific to a defined set of scenarios and transactions supported by relevant Oracle Fusion Middleware 14*c* product components. It applies where validated cryptography is used to support or enforce security-sensitive tasks such as authentication, authorization, confidentiality, integrity, and so on.

The use of cryptographic services for other tasks that are non-security sensitive does not require FIPS 140 compliance. Oracle Fusion Middleware 14*c* supports enabling FIPS 140 mode for security-sensitive scenarios while complying and co-existing with product functionality that does not require operating in that mode.

- About FIPS 140-2 Validated Libraries
  Oracle Fusion Middleware 14*c* includes FIPS 140-validated RSA libraries from RSA, the Security Division of EMC (RSA) to support FIPS 140 operation. Algorithms not approved under FIPS 140 are disabled within the RSA libraries.
- About Provider and Algorithm Selection
  FIPS 140 implementation in Oracle Fusion Middleware occurs in the context of the Java platform's Java Cryptography Architecture (JCA). To accommodate the co-existence of FIPS 140-validated algorithms for security-sensitive tasks as well as algorithms for other tasks, additional cryptographic providers are also configured to provide functionality not supported in FIPS 140-validated RSA libraries, and for certain non-compliant cryptographic functions such as MD5, which are disabled within the FIPS 140-validated RSA libraries.

About FIPS 140-2 Validated Libraries

Oracle Fusion Middleware 14*c* includes FIPS 140-validated RSA libraries from RSA, the Security Division of EMC (RSA) to support FIPS 140 operation. Algorithms not approved under FIPS 140 are disabled within the RSA libraries.

The libraries are based on RSA version 6.2 BSAFE and JCE software and include the following modules:

- Crypto-J V6.2.4.0.1
- SSL-J V6.2.4
- Cert-J V6.2.4

> **Note:**
>
> The April 2021 Patch Set Update (PSU) adds support for:
>
> - Crypto-J V6.2.5
> - SSL-J V6.2.6
> - Cert-J V6.2.4.0.1

In addition to the continued support for RSA keys, Oracle Fusion Middleware 14*c* also supports Elliptic Curve Cryptography (ECC). ECC is emerging as an attractive public-key cryptography

because it offers equivalent security with smaller key sizes, which results in faster computations, lower power consumption, and memory and bandwidth savings.

> **Note:**
>
> These are the FIPS 140-certified library and module versions at the time of publication. The actual versions in effect at your installation could be slightly different from the ones listed here, as the vendor may issue some patches between certification and the time the product actually shipped. Thus the actual version could be a dot release of the certified version.
>
> The version number is for information only; you can do any independent verification of certification and strength of algorithms.

For background about the FIPS 140 standards and algorithms, refer to the FIPS 140-2 documentation at:

`http://csrc.nist.gov/publications/PubsFIPS.html`

About Provider and Algorithm Selection

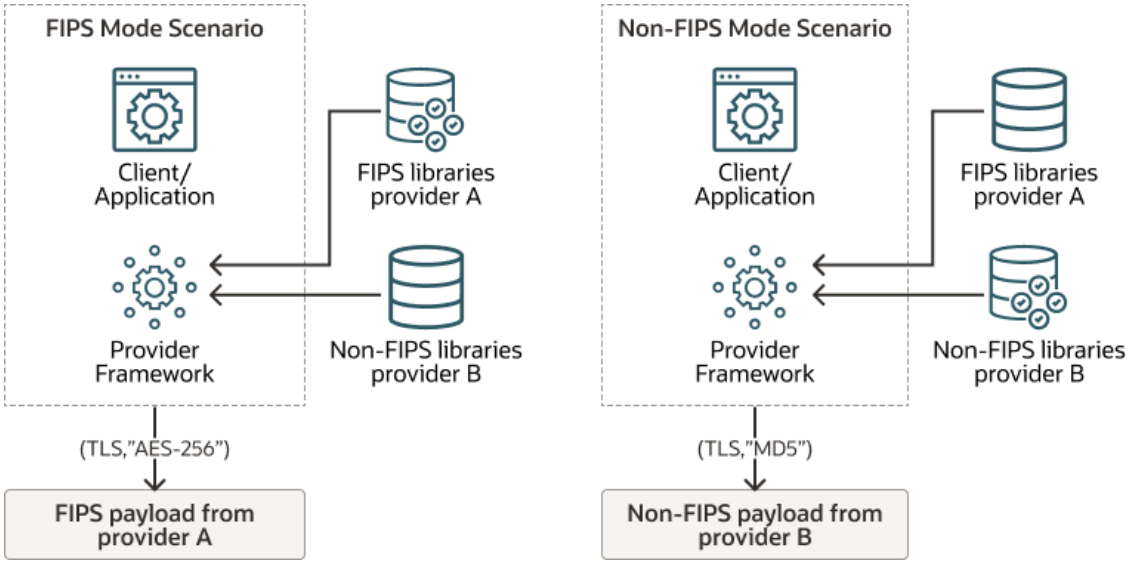
FIPS 140 implementation in Oracle Fusion Middleware occurs in the context of the Java platform's Java Cryptography Architecture (JCA). To accommodate the co-existence of FIPS 140-validated algorithms for security-sensitive tasks as well as algorithms for other tasks, additional cryptographic providers are also configured to provide functionality not supported in FIPS 140-validated RSA libraries, and for certain non-compliant cryptographic functions such as MD5, which are disabled within the FIPS 140-validated RSA libraries.

The basic flow is as follows:

- An application (for example, an external web client or Oracle HTTP Server) requests a service or connection to a server such as WebLogic Server. The request typically involves a "payload" such as a data packet to be transmitted.
- JCA evaluates the request to determine whether FIPS 140 compliance is required.
- The request is routed to JCA's "provider" framework, which contains a set of (FIPS 140-compliant and non-compliant) providers for digital signatures, message digests (hashes), certificates, and certificate validation, encryption, and other cryptographic services.
- The providers are searched in preference order and the implementation from the first provider that supplies the correct algorithm is returned. For the security-sensitive cases, only FIPS 140 compliant algorithms are used to execute the cryptographic operations.

Figure 8-1 illustrates this flow:

Figure 8-1 Selecting a FIPS 140 Provider



- The first request, on the left, is made in a security-sensitive scenario. JCA uses the SHA-256 provider from the RSA cryptographic library to process the request and deliver the FIPS 140 payload.
- The second request, on the right, is executed in a non-sensitive scenario. JCA uses the MD5 provider from the non-cryptographic library to process the request with the non-FIPS 140 payload.

Thus, a security-sensitive scenario such as HTTPS/TLS inbound and outbound communication which is intended to be FIPS 140-compliant uses only those cryptographic functions available in the FIPS 140-validated RSA libraries to encrypt and sign HTTPS/TLS network payloads.

Components with FIPS 140 Support

When you plan to work with FIPS 140 in Oracle Fusion Middleware, be aware of the different components at various layers of the middleware stack where certain features may operate in FIPS 140 mode. If any component in the stack is operating in non-FIPS 140 mode, the transaction may not be FIPS 140-compliant. It is therefore important to ensure that all relevant components are operating in FIPS 140 mode.

Table 8-1 lists the components where you can enable FIPS 140, and contains the following details:

- The Oracle Fusion Middleware layer where the component resides;
- the component name
- the scenario which can be FIPS 140-enabled
- cross-reference to product documentation for details, including how to enable or disable FIPS 140, other relevant configuration details, and what product functions support the use of FIPS 140-validated cryptography.

> **Note:**
>
> Not all features of each listed component are FIPS 140-compliant. Only the specified features support FIPS 140.

Table 8-1 Components with FIPS 140-2 Support in Oracle Fusion Middleware

Component Layer	Component	Feature	Details
Fusion Middleware Core	Oracle HTTP Server	• TLS Inbound (HTTPS) • TLS Outbound from OHS to any web, proxy or application server using OHS SSL proxy (mod_proxy, mod_ossl) *Note*: For outbound connections from OHS to WLS, FIPS must be enabled at WebLogic (for inbound connections) to enable FIPS communication between OHS and WLS.	These topics in *Administering Oracle HTTP Server*: • "SSLFIPS" directive for mod_ossl • "Managing Application Security"
Fusion Middleware Core	Oracle WebLogic Server	• TLS inbound: HTTPS, T3S, JMX/T3S, JMS • TLS outbound: HTTPS, T3S, JMX/T3S, JMS, JDBC (Oracle RDBMS) • Database Connections (through Data Source)	"Enabling FIPS Mode" in *Administering Security for Oracle WebLogic Server* "Use the SHA-256 Secure Hash Algorithm" in *Securing WebLogic Web Services for Oracle WebLogic Server* "Using Encrypted Connection Properties" in *Administering JDBC Data Sources for Oracle WebLogic Server*
Fusion Middleware Core	Oracle Platform Security Services	• Keystore Service • Credential Store Service	"FIPS Support in OPSS" in *Securing Applications with Oracle Platform Security Services*
Fusion Middleware Core	Oracle Web Services Manager	• Message protection • Token signature	"Supported Algorithm Suites" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*
Fusion Middleware Core	Oracle SOA Suite	• JCA Adapter for Files/FTP • JCA Adapter for Database • JCA Adapter for JMS • Service Bus	"About FIPS Compliance for the SFTP Transport" in *Developing Services with Oracle Service Bus* "Enabling FIPS Compliance in Oracle File and FTP Adapters" in *Understanding Technology Adapters*
Fusion Middleware Core	Oracle Identity Manager	• Changing Client and Service webpolicies	Changing Client Policies to Create Custom Policy for FIPS in *Administering Oracle Identity Governance*.
Database	Oracle Database	• Database in FIPS 140-2 mode	"Oracle Database FIPS 140–2 Settings" in *Oracle Database Security Guide*

> **Note:**
>
> Database is included for reference. Consult the certification matrix for supported versions and other details.
>
> For detailed information about SSL FIPS 140-2 for OHS, OWLS, OPSS, and OWSM, refer to support Document 2115681.1 on My Oracle Support. You can access My Oracle Support at: https://support.oracle.com/.

Common Scenarios for an Operational FIPS 140-2 Environment

The implementation of a feature and the establishment of a connection between a client and a server are some of the possible scenarios you may use or encounter while operating in a FIPS 140–2 environment. Each component scenario uses a corresponding communication protocol and signature algorithm.

Table 8-1 listed the components in Oracle Fusion Middleware with FIPS 140-2 features. Table 8-2 lists typical protocols for each component scenario:

> **Note:**
>
> These are representative scenarios - the table is not intended to provide a comprehensive listing of all possible scenarios.

Table 8-2 FIPS 140-2 Scenarios

Feature or Connection	Communication Protocol	Signature Algorithm/Protocol Details
Inbound connection from an external web client or application to Oracle HTTP Server	• HTTPS (Client Access to OHS) • SOAP-TLS (Server to Server Communication)	HTTPS Server (TLS, Mutual Authentication, RSA-2048 with SHA-256 X.509 Certificates, AES-256 Bulk Data Encryption, ECDSA Signing Algorithm and ECDH Key Agreement)
Outbound connection from Oracle HTTP Server to Oracle WebLogic Server	• HTTPS (OHS to HTTP Servlet in WLS) for end-end SSL with external SSL termination in OHS.	HTTPS Client (TLS, Mutual Authentication, RSA-2048 with SHA-256 X.509 Certificates, AES-256 Bulk Data Encryption, ECDSA Signing Algorithm and ECDH Key Agreement)
Inbound connection from an external web client or application to Oracle WebLogic Server	• HTTPS (Client Access to HTTP Servlet) • SOAP-TLS (Server to Server Communication)	HTTPS Server (TLS, Mutual Authentication, RSA-2048 with SHA-256 X.509 Certificates, AES-256 Bulk Data Encryption)
Outbound connection from Oracle WebLogic Server to an external web, proxy or application server	• HTTPS (WLS to an external HTTPS server) • SOAP-TLS (Server to Server Communication)	HTTPS Client (TLS, Mutual Authentication, RSA-2048 with SHA-256 X.509 Certificates, AES-256 Bulk Data Encryption)
Outbound connection from Oracle WebLogic Server to Oracle Database 11gR2	• DB-TLS-jdbc (WebLogic to Database Communication)	JDBC (TLS, Mutual Authentication, RSA-2048 with SHA-256 X.509 Certificates, AES-256 Bulk Data Encryption)

Table 8-2 (Cont.) FIPS 140-2 Scenarios

Feature or Connection	Communication Protocol	Signature Algorithm/Protocol Details
XML Message Protection (XML Signing) for SOAP messages using Oracle Web Services Manager	• SOAP-MsgSec	XML Signature (RSA-SHA256, HMAC-SHA256); Entire Body, Include SwA Attachment
XML Message Protection (XML Encryption) for SOAP messages using Oracle Web Services Manager	• SOAP-MsgSec	XML Signature (RSA-SHA256, HMAC-SHA256); Entire Body, Include SwA Attachment
Inbound JMS connection to Oracle WebLogic Server	• JMS traffic is secure in flight	JMS/TLS
Outbound JMS connection from Oracle WebLogic Server	• JMS traffic is secure in flight	JMS/TLS
Secure JNDI lookups from deployed components	• JDNI-EJB	T3S
Secure administrator access to servers	• WLST traffic to WLS server is secure in flight	T3S
Keystore and Certificate Generation	• Encryption • Key Exchange	RSA 2048, AES 256, SHA-2
Hashing Algorithms, Password-Based Encryption	• Hashing • Encryption	SHA-2
Oracle Service Bus for SOA service-based components	SFTP transport for service types: • Messaging • Any XML	Public Key Algorithm (diffie-hellman-group14-sha1) Key Exchange Algorithm (ssh-rsa)
Managed File Transfer (MFT) • Key exchange • Ciphers • Message Authentication	File transports: • SFTP • FTP-SSL • PGP • JCA Transport	Typical algorithms • DHG14 • AES128CBC, TripleDESCBC • HMACSHA1 • RSA, DSA • Diffie-hellman-group14-sha1
JCA Adapters	File Transfer Protocol	• diffie-hellman-group14-sha1 • ssh-rsa

> **Note:**
>
> Unless otherwise indicated, all component servers are at Release 12*c* (12.2.1.2).

Troubleshooting FIPS 140 Issues

You may encounter problems while configuring FIPS 140 for different components of the Oracle Fusion Middleware and need information on how to troubleshoot those problems.

This section explains how to troubleshoot issues encountered with FIPS 140 configuration. It contains these topics:

- FIPS 140 Troubleshooting for Stand-alone WebLogic Server
  Follow these steps to troubleshoot FIPS 140 mode for a stand-alone Oracle WebLogic Server while configuring a WebLogic Server and the Data Source properties.
- FIPS 140 Troubleshooting for Oracle Platform Security Services
  Find out how to troubleshoot issues that originate at different stages of FIPS configuration in Oracle Platform Security Services (OPSS).
- FIPS 140 Troubleshooting for Oracle Web Services Manager
  Find out how to troubleshoot issues originating in Oracle Web Services Manager during message protection policy enforcement.
- FIPS 140 Troubleshooting for Database and JDBC Driver
  Review this topic for information about security configuration for the database, the JDBC driver, including data source issues related to database.

FIPS 140 Troubleshooting for Stand-alone WebLogic Server

Follow these steps to troubleshoot FIPS 140 mode for a stand-alone Oracle WebLogic Server while configuring a WebLogic Server and the Data Source properties.

During WebLogic Server Configuration

1. Make sure to prepend the server CLASSPATH with jcmFIPS.jar and sslj.jar.
2. To explicitly verify *`AES_256*` cipher suites, update the local_policy.jar and US_export_policy.jar in the JAVA_HOME/jre/lib/security directory with the corresponding file with unlimited strength.
3. Modify JAVA_HOME/jre/lib/security/java.security by putting security.provider.1=com.rsa.jsafe.provider.JsafeJCE and security.provider.2=com.rsa.jsse.JsseProvider on top of the list.

For more information on FIPS mode in Oracle WebLogic Server, see "Enabling FIPS Mode" in *Administering Security for Oracle WebLogic Server*.

During Data Source Configuration

Make sure that the value of the DataSource property `oracle.net.ssl_version` is set to **1.0**.

> **Note:**
>
> `oracle.net.ssl_version` is an optional Oracle WebLogic Server DataSource configuration property. A value of 1.0 represents connection through TLS v 1.0 Protocol.

FIPS 140 Troubleshooting for Oracle Platform Security Services

Find out how to troubleshoot issues that originate at different stages of FIPS configuration in Oracle Platform Security Services (OPSS).

During WebLogic Domain Creation

You may see the following exceptions in `wlsconfig_xxxxx.log` during domain creation in FIPS 140 mode:

```
"CFGFWK-60455: The password
must be at least 8 alphanumeric characters with at least one number or
special character."

"Caused by: java.lang.NoSuchMethodError:
com.rsa.jsafe.JSAFE_SecretKey.generateInit([ILjava/security/SecureRandom;)"
```

This exception may occur if you are using cryptoJ 5 jars. Make sure you have installed Oracle WebLogic Server with cryptoJ 6 jars to avoid this error.

When Exporting from Domain Keystore

If you are using JKS and JCEKS type keystores in a FIPS 140-enabled domain, and see the following error:

```
Command FAILED, Reason:
 oracle.security.jps.service.keystore.KeyStoreServiceException: Failed to export the
keystore
```

make sure that you have configured the following providers in the `java.security` file:

```
sun.security.provider.Sun
com.sun.crypto.provider.SunJCE
```

During Key or Certificate Generation

When generating key or certificate with password protection, you may get the following error:

```
javax.management.MBeanException: javax.management.MBeanException:
oracle.security.jps.service.keystore.KeyStoreServiceException: Failed to generate CA
signed certificate.
```

make sure that you use permission protection.

FIPS 140 Troubleshooting for Oracle Web Services Manager

Find out how to troubleshoot issues originating in Oracle Web Services Manager during message protection policy enforcement.

During Message Protection Policy Enforcement

If you see this error during Oracle Web Services Manager message protection policy enforcement:

```
Caused by: java.lang.SecurityException: Algorithm not allowable in FIPS140 mode: MD5
       at com.rsa.cryptoj.o.cc.b(Unknown Source)
       at com.rsa.cryptoj.o.cc.f(Unknown Source)
```

make sure that certificates used in message protection enforcement are generated using FIPS 140-compliant algorithms like SHA1WithRSA or SHA256WithRSA.

If you encounter this error for the JKS keystore during message protection policy enforcement:

```
oracle.fabric.common.PolicyEnforcementException: WSM-00143 : Failure creating Java
Keystore instance for type JKS.
```

make sure that `sun.security.provider.Sun` is configured in the JDK.

FIPS 140 Troubleshooting for Database and JDBC Driver

Review this topic for information about security configuration for the database, the JDBC driver, including data source issues related to database.

For complete details, see the white paper "SSL With Oracle JDBC Thin Driver" on the Oracle Technology Network at:

http://www.oracle.com/technetwork/middleware/id-mgmt/documentation/index.html

Part IV
Deploying Applications

You should understand the deployment process and how to deploy applications to Oracle Fusion Middleware

9

Understanding the Deployment Process

Before you deploy Oracle Fusion Middleware applications, such as Jakarta EE applications, you should understand the deployment process, such as designing and developing applications and deploying those applications to Managed Servers.

- What Is a Deployer?
  A user in the role of **deployer** is responsible for deploying applications, such as Jakarta EE applications, and ADF applications, to WebLogic Server instances or clusters.
- General Procedures for Moving from Application Design to Production Deployment
  You design and test your application with the integrated Oracle WebLogic Server. Then, you can deploy the application to a Managed Server. You can move from application design and development to deployment in a production environment.
- Diagnosing Typical Deployment Problems
  If you encounter problems when you deploy an application, you can diagnose those problems and correct them.

What Is a Deployer?

A user in the role of **deployer** is responsible for deploying applications, such as Jakarta EE applications, and ADF applications, to WebLogic Server instances or clusters.

A user who is functioning as a deployer should be granted the Oracle WebLogic Server deployer security role. The deployer security role allows deployment operations, as well as viewing the server configuration and changing startup and shutdown classes. To grant this role to a user, from Fusion Middleware Control:

1. From the **WebLogic Domain** menu, select **Security,** then **Users and Groups.**

   The Users and Groups page is displayed.
2. If you do not have such a user, click **Create.**

   The Create a User page is displayed.
3. Enter a name for the user and a password. Confirm the password.
4. Click **Create.**
5. On the Users and Groups page, select the user.
6. On the Settings for User page, select **Deployers** from the **Available** pane and move it to the **Chosen** pane.
7. Click **Save.**

General Procedures for Moving from Application Design to Production Deployment

You design and test your application with the integrated Oracle WebLogic Server. Then, you can deploy the application to a Managed Server. You can move from application design and development to deployment in a production environment.

Designing and Developing an Application

In many cases, developers use Oracle JDeveloper to create their applications. Oracle JDeveloper is an integrated development environment (IDE) for building service-oriented applications using the latest industry standards for Java, XML, Web services, portlets, and SQL. JDeveloper supports the complete software development life cycle, with integrated features for modeling, coding, debugging, testing, profiling, tuning, and deploying applications.

In this environment, you use the integrated Oracle WebLogic Server, which is packaged with Oracle JDeveloper for testing your applications.

For information about developing your applications, see:

- *Developing Applications for Oracle WebLogic Server*
- *Developing Fusion Web Applications with Oracle Application Development Framework*
- *Developing SOA Applications with Oracle SOA Suite*
- *Developing for Oracle WebCenter Portal*

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 Jakarta EE applications, see *Deploying Applications to Oracle WebLogic Server*
- For Oracle ADF, see *Administering Oracle ADF Applications*
- For Oracle SOA Suite, see *Developing SOA Applications with Oracle SOA Suite*
- For Oracle WebCenter Portal, see *Administering Oracle WebCenter Portal*

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 Jakarta EE applications, you package the application in an EAR file. See Preparing Applications and Modules for Deployment in *Deploying Applications to Oracle WebLogic Server*.
   - For Oracle ADF, you package the application in an EAR file. See What You May Need to Know About EAR Files and Packagingin *Developing Fusion Web Applications with Oracle Application Development Framework*.
   - For Oracle SOA Suite, you package the application into a JAR or ZIP file. See Understanding the Packaging Impact in *Developing SOA Applications with Oracle SOA Suite*.

- For Oracle WebCenter Portal, you package the application in an EAR file. See *Developing for Oracle WebCenter Portal*

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 Extending a Domain to Support Additional Components.

     For more information about installing and configuring specific components, see:

     – For Oracle ADF: How to Install the ADF Runtime to the WebLogic Installation in *Administering Oracle ADF Applications*

     – For Oracle SOA Suite: Installing Oracle SOA Suite and Oracle Business Process Management Suite, Configuring the Oracle SOA Suite Domain and Configuring the Oracle Business Process Management Domain in *Installing and Configuring Oracle SOA Suite and Business Process Management*

     – For Oracle WebCenter Portal: Installing Oracle WebCenter Portal and Configuring Oracle WebCenter Portal in *Installing and Configuring Oracle WebCenter Portal*
   - Creating any necessary schemas in an existing database. See *Creating Schemas with the Repository Creation Utility*.
   - Registering the MDS Repository with the Oracle WebLogic Server domain, if your application uses the MDS Repository. For example, Oracle SOA Suite applications require MDS. Some ADF applications involve customizations using MDS. See Registering a Database-Based MDS Repository Using Fusion Middleware Control 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 Jakarta EE applications: *Administering JDBC Data Sources for Oracle WebLogic Server*
   - For Oracle ADF: How to Create a JDBC Data Source for Oracle WebLogic Server in *Administering Oracle ADF Applications*
   - For Oracle SOA Suite: Creating Data Sources and Queues in *Developing SOA Applications with Oracle SOA Suite*
   - For Oracle WebCenter Portal: See *Administering Oracle WebCenter Portal*
4. For Oracle SOA Suite, create connection factories and connection pooling. See Creating Connection Factories and Connection Pooling in *Developing SOA Applications with 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 about creating connections, see:
   - For Oracle ADF: How to Create a Connection to the Target Application Server in *Developing Fusion Web Applications with Oracle Application Development Framework*
   - For Oracle SOA Suite: Creating an Application Server Connection in *Developing SOA Applications with Oracle SOA Suite*

- For Oracle WebCenter Portal: See *Developing for Oracle WebCenter Portal*

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 *Developing SOA Applications with 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. See:
   - For pure Jakarta EE applications: Creating a New Deployment Plan to Configure an Application in *Deploying Applications to Oracle WebLogic Server*
   - For Oracle SOA Suite: Introduction to Configuration Plans in *Developing SOA Applications with Oracle SOA Suite*
8. Migrate application security, such as credentials, identities, and policies. See:
   - For pure Jakarta EE applications: Migrating Security Data in *Administering Security for Oracle WebLogic Server*
   - For Oracle ADF: Preparing the Secure Application for Deployment in *Developing Fusion Web Applications with Oracle Application Development Framework*
   - For Oracle SOA Suite: Enabling Security in *Developing SOA Applications with Oracle SOA Suite*
   - For Oracle WebCenter Portal: Managing WebCenter Portal Application Security in *Administering Oracle WebCenter Portal*
9. Create a deployment profile. A **deployment profile** packages or archives a custom ADF, WebCenter Portal, 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 the project and application level.

   For more information about deployment profiles, see:
   - For Oracle ADF: How to Create Deployment Profiles in *Developing Fusion Web Applications with Oracle Application Development Framework*
   - For Oracle SOA Suite: Optionally Creating a Project Deployment Profile in *Developing SOA Applications with Oracle SOA Suite*
   - For Oracle WebCenter Portal: See *Developing for Oracle WebCenter Portal*
10. Migrate Oracle JDeveloper extensions for Oracle SOA Suite. Table 9-1 shows the extensions and where they are documented:

Table 9-1 Oracle JDeveloper Extensions

Component	Extension	See:
Oracle SOA Suite	SOA extensions	Enabling Oracle JDeveloper Extensions in *Installing Oracle JDeveloper*
Oracle WebCenter Portal	WebCenter Portal extensions	*Developing for Oracle WebCenter Portal*

11. Deploy the application to a Managed Server.

    See:
    - For pure Jakarta EE applications: Exporting an Application for Deployment to New Environments in *Deploying Applications to Oracle WebLogic Server*

- For Oracle ADF: Deploying the Application in *Developing Fusion Web Applications with Oracle Application Development Framework*
- For Oracle SOA Suite: Deploying SOA Composite Applications in *Developing SOA Applications with Oracle SOA Suite*
- For Oracle WebCenter Portal: See *Administering Oracle WebCenter Portal*

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 Jakarta EE applications: Using the WebLogic Scripting Tool in *Understanding the WebLogic Scripting Tool*
- For Oracle ADF: Deploying Using Scripts and Ant in *Administering Oracle ADF Applications*
- For Oracle SOA Suite: The following sections in *Developing SOA Applications with Oracle SOA Suite*:
  - Managing SOA Composite Applications with Scripts
  - Managing SOA Composite Applications with ant Scripts
- For Oracle WebCenter Portal: See *Developing for Oracle WebCenter Portal*

Diagnosing Typical Deployment Problems

If you encounter problems when you deploy an application, you can diagnose those problems and correct them.

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 Deploying an Application to Managed Servers.
- Oracle JDeveloper extensions. Ensure that you have migrated any Oracle JDeveloper extensions. See Table 9-1.
- Data sources. Ensure that you have correctly configured JDBC data sources. See Step 3 in Deploying an Application to Managed Servers.
- Security configuration. Ensure that you have migrated application security, such as credentials, identities, and policies. See Step 8 in Deploying an Application to Managed Servers.

In addition, see Troubleshooting Common Deployment Errors in *Developing SOA Applications with Oracle SOA Suite* for information about troubleshooting SOA applications.

10

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, redeploy, and undeploy applications to Oracle Fusion Middleware.

- Overview of Deploying Applications
  Oracle WebLogic Server provides a Jakarta EE-compliant infrastructure for deploying, undeploying, and redeploying Jakarta EE-compliant applications and modules.
- Understanding and Managing Data Sources
  A **data source** is a Java object that application components use to obtain connections to a relational database.
- Deploying, Undeploying, and Redeploying Jakarta EE Applications
  A Jakarta EE application consists of one or more components, which can be application clients, web components, and EJB components, and can be deployed on Manager Servers.
- Deploying, Undeploying, and Redeploying Oracle ADF Applications
  Oracle ADF is an end-to-end application framework that builds on Java Platform, Enterprise Edition (Jakarta EE) standards and open-source technologies to simplify and accelerate implementing service-oriented applications.
- Deploying, Undeploying, and Redeploying SOA Composite Applications
  Oracle SOA Suite uses the SCA standard as a way to assemble service components into a SOA composite application. You can deploy, undeploy, and redeploy SOA composite applications.
- Deploying, Undeploying, and Redeploying WebCenter Portal Applications
  Oracle WebCenter Portal applications differ from traditional Jakarta 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.
- Managing Deployment Plans
  A **deployment plan** is a client-side aggregation of all the configuration data needed to deploy an archive into Oracle WebLogic Server. A deployment plan allows you to easily deploy or redeploy an application using a saved set of configuration settings.
- About the Common Deployment Tasks in Fusion Middleware Control
  When you deploy an application using Fusion Middleware Control, you can use the Deployment Settings page of the Deployment wizard to perform specific deployment configuration tasks before the application is deployed.
- Configuring Applications in Fusion Middleware Control
  With Fusion Middleware Control, you can configure attributes of your application.
- Changing MDS Configuration Attributes for Deployed Applications
  If your application uses an MDS Repository, you can modify configuration attributes after the application is deployed.

Overview of Deploying Applications

Oracle WebLogic Server provides a Jakarta EE-compliant infrastructure for deploying, undeploying, and redeploying Jakarta EE-compliant applications and modules.

What Types of Applications Can You Deploy?

You can deploy the following into Oracle WebLogic Server:

- A complete Jakarta EE application packaged as an Enterprise Archive (EAR) file.
- Standalone modules packaged as Java Archive files (JARs) containing Web services, Jakarta Enterprise Beans (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 (Jakarta 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 Portal application. WebCenter Portal applications differ from traditional Jakarta 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 Portal applications.

> **Note:**
>
> If your application uses password indirection in the application-level data source, you cannot use Fusion Middleware Control to deploy the application. The section "Deploying an Application to an EAR File to run on Oracle WebLogic Server" in the Oracle JDeveloper Help describes how to change the settings of the application to be able to deploy the application using Fusion Middleware Control.

You can use Fusion Middleware Control, Oracle WebLogic Remote 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 10-1.

Table 10-1 Tools to Deploy Applications

Type of Application	Tools to Use
Pure Jakarta EE application	Oracle WebLogic Remote Console Fusion Middleware Control: Deployment Wizard Oracle JDeveloper WLST command line
ADF application	Fusion Middleware Control: Deployment Wizard Oracle JDeveloper WLST command line

Table 10-1 (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 Portal 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 Jakarta 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 Managing the MDS Repository.

> **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 *Securing Applications with Oracle Platform Security Services*.

About Deployment, Redeployment, and Undeployment

When you deploy an application, you deploy it to the application server for the first time.

When you redeploy an application, you can:

- Redeploy a new version of the application; the previous version is still available, but the state is set to "Retired."

  This is known as the production redeployment strategy. Oracle WebLogic Server automatically manages client connections so that only new client requests are directed to the new version. Clients already connected to the application during the redeployment continue to use the older version of the application until they complete their work, at which point Oracle WebLogic Server automatically retires the older application.
- Redeploy the same version of the application or redeploy an application that is not assigned a version; the application version you select is replaced with the new deployment.
- Redeploy a previous version of the application; the earlier, retired version is set to "Active" and the later version is set to "Retired."

When you undeploy an application, Oracle WebLogic Server stops the application and removes staged files from target servers. It does not remove the original source files used for deployment.

Understanding and Managing Data Sources

A **data source** is a Java object that application components use to obtain connections to a relational database.

The following topics describe data sources and how to manage them:

About Data Sources

A **data source** is a Java object that application components use to obtain connections to a relational database. Specific connection information, such as the URL or user name and password, are set on a data source object as properties and do not need to be explicitly defined in an application's code. This abstraction allows applications to be built in a portable manner, because the application is not tied to a specific back-end database. The database can change without affecting the application code.

Applications use the Java Naming and Directory Interface (JNDI) API to access a data source object. The application uses a JNDI name that is bound to the data source object. The JNDI name is logical and can be mapped to any data source object. Like data source properties, using JNDI provides a level of abstraction, since the underlying data source object can change without any changes required in the application code. The result is that the details of accessing a database are transparent to the application.

See Administering JDBC Data Sources for Oracle WebLogic Server for more information about data sources.

When you configure certain Oracle Fusion Middleware components, such as Oracle SOA Suite, using the Oracle WebLogic Server Configuration Wizard, you specify the data source connection information. If the components use the MDS Repository, the Configuration Wizard prepends `mds-` to the data source name to indicate that the data source is a system data source used by MDS Repository.

See *Creating WebLogic Domains Using the Configuration Wizard* for information about specifying data sources with the Configuration Wizard.

If you are using Oracle Real Application Clusters (Oracle RAC) or Oracle Fusion Middleware Cold Failover Cluster, you must configure one of the following types of data sources:

- **Multi data sources**

  To use multi data sources, you must use the Oracle WebLogic Remote 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 Remote Console. For example, the MDS Repository is not listed in the Fusion Middleware Control navigation pane and is not displayed as a choice for a target metadata repository when you deploy an application.

- **GridLink data sources**

  To use GridLink data sources, you can use the Oracle WebLogic Remote Console or Fusion Middleware Control, as described in Creating a GridLink Data Source Using Fusion Middleware Control.

See Administering JDBC Data Sources for Oracle WebLogic Server for more information about configuring multi data sources and GridLink data sources.

Creating and Managing JDBC Data Sources

You can create and manage JDBC data sources using the following management tools:

- The Oracle WebLogic Remote Console
- The WebLogic Scripting Tool (WLST)
- Fusion Middleware Control

To create an MDS data source manually, you should use Fusion Middleware Control or WLST to set the correct attributes for the data source. The MDS data source is displayed in the navigation pane in Fusion Middleware Control. 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 Managing the MDS Repository.

> **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 Remote Console to create a MDS data source. If you do, note the following:

- You must prefix the data source name with `mds-` if you intend it to be used with MDS Repository.
- You must target the data source to the Administration Server and to all Managed Servers to which you are deploying applications that need the data source.
- You must turn off global transactions.

See Administering JDBC Data Sources for Oracle WebLogic Server for information about creating and managing a data source using the Oracle WebLogic Remote Console or WLST and for more information about configuring multiple data sources.

The following topics describe how to create and manage JDBC data sources with Fusion Middleware Control:

- Creating a JDBC Data Source Using Fusion Middleware Control
- Editing a JDBC Data Source Using Fusion Middleware Control
- Monitoring a JDBC Data Source Using Fusion Middleware Control
- Controlling a JDBC Data Source Using Fusion Middleware Control
- Creating a GridLink Data Source Using Fusion Middleware Control

Creating a JDBC Data Source Using Fusion Middleware Control

To create a JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources.**

   The JDBC Data Sources page is displayed.

2. From **Create,** select **Generic Data Source.**
3. Follow the instructions in the wizard to set the properties of the data source and to target the data source for one or more of the Managed Servers in the domain.

   For help on individual fields and properties, use your mouse to give focus to a field. Fusion Middleware Control displays a popup definition of the field.

   Note that the data source properties you define in Fusion Middleware Control are similar to those you define when creating data sources in the Oracle WebLogic Remote Console. As a result, you can also refer to Creating a JDBC Data Source in *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about the data source properties.

Editing a JDBC Data Source Using Fusion Middleware Control

To edit an existing JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

   The JDBC Data Sources page is displayed.
2. Click the data source that you want to edit.

   The page for that particular JDBC Data Source is displayed.
3. Use the tabs on this page to modify the properties of the selected data source.

   For help on individual fields and properties, use your mouse to give focus to a field. Fusion Middleware Control displays a popup definition of the field.

   Note that the data source properties you edit in Fusion Middleware Control are similar to those you edit when editing data sources in the Oracle WebLogic Remote Console. As a result, you can also refer to Creating a JDBC Data Source in *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about the data source properties.

Monitoring a JDBC Data Source Using Fusion Middleware Control

To monitor a JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

   The JDBC Data Sources page is displayed.
2. Select the data source that you want to monitor.
3. Select the Monitoring tab to display the statistics for the JDBC data source.

   This page shows the current instances of the selected data source.

   Note that only data sources that are targeted to a running Managed Server are shown on this page. If a specific data source is not listed on the monitoring page, then edit the data source to be sure it is targeted to a running Managed Server.
4. For each data source instance, review the performance metrics.

Controlling a JDBC Data Source Using Fusion Middleware Control

To start, stop, suspend, resume, or clear the statement cache for a JDBC data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

The JDBC Data Sources page is displayed.

2. Select the data source that you want to control
3. Select the Control tab.

   Note that only data sources that are targeted to a running Managed Server are shown on this page. If a specific data source is not listed on the control page, edit the data source to be sure that it is targeted to a running Managed Server.
4. Select the instance and click Start, Stop, Resume, Suspend, Shrink, Reset, or Clear Statement Cache to control or change the state of the selected JDBC data source.

Creating a GridLink Data Source Using Fusion Middleware Control

A single GridLink data source provides connectivity between WebLogic Server and an Oracle Database service targeted to an Oracle RAC cluster. For detailed information about GridLink data sources, see Creating a GridLink Data Source in *Administering JDBC Data Sources for Oracle WebLogic Server*.

To create a Grid Link data source using Fusion Middleware Control:

1. From the **WebLogic Domain** menu, choose **JDBC Data Sources**.

   The JDBC Data Sources page is displayed.
2. From **Create,** select **GridLink Data Source.**
3. Follow the instructions in the wizard to set the properties of the data source and to target the data source for one or more of the Managed Servers in the domain.

   For help on individual fields and properties, use your mouse to give focus to a field. Fusion Middleware Control displays a popup definition of the field.

   Note that the data source properties you define in Fusion Middleware Control are similar to those you define when creating data sources in the Oracle WebLogic Remote Console. As a result, you can also refer to Creating a GridLink Data Source in *Administering JDBC Data Sources for Oracle WebLogic Server* for more information about the data source properties.

Deploying, Undeploying, and Redeploying Jakarta EE Applications

A Jakarta EE application consists of one or more components, which can be application clients, web components, and EJB components, and can be deployed on Manager Servers.

You can use Fusion Middleware Control, Oracle WebLogic Remote Console, Oracle JDeveloper, or the command line to deploy, undeploy, or redeploy a Jakarta EE application.

The following topics describe using Fusion Middleware Control and the command line to accomplish these tasks:

Deploying Applications to Oracle WebLogic Server describes deploying applications using Oracle WebLogic Remote Console and the WLST command line.

- About Managed Server Independence and Deploying Applications
  Managed Server Independence for Deployment (MSI-D) lets administrators update applications and libraries without requiring contact with the Administration Server. MSI-D makes the deployment of applications and libraries more resilient to resource and network unavailability.

About Managed Server Independence and Deploying Applications

Managed Server Independence for Deployment (MSI-D) lets administrators update applications and libraries without requiring contact with the Administration Server. MSI-D makes the deployment of applications and libraries more resilient to resource and network unavailability.

MSI-D lets you upgrade or patch applications and libraries by simply placing the new application or library version in a predefined directory without invoking any deployment commands; this is also referred to as hot patching.

Using the WebLogic Remote Console and Fusion Middleware Control, you can view, test, and monitor MSI-D applications and libraries. However, the typical deployment and control operations (such as start, stop, update, and delete) are not supported for these applications and libraries.

For more information, see Understanding Managed Server Independence Mode in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

Deploying Jakarta EE Applications

You can deploy an application to a Managed Server instance or a cluster. This section describes how to deploy an application to a Managed Server. It contains the following topics:

Deploying Jakarta EE Applications Using Fusion Middleware Control

To deploy a Jakarta EE application to a Managed Server using Fusion Middleware Control:

1. From theWebLogic Domain menu, select **Control,** then **Deployments.**

   The Deployments page is displayed.

2. From the Deployments menu, select **Deploy** to open the Deploy Jakarta EE Application Assistant.

   The Select Archive page is displayed.

3. 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.
4. 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.

5. In the Deployment Type section, you can select one of the following:
   - **Deploy this archive or exploded directory as an application**
   - **Deploy this archive or exploded directory as a library**
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 Remote 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:
    - **Install and start application (servicing all requests)**
    - **Install and start application in administration mode (servicing only admin requests)**
    - **Install only. Do not start**
12. You can expand Other Options, which provides the following for Application Source Accessibility:
    - Use the defaults defined by the deployment's targets. Recommended selection.
    - Copy this application onto every target. During deployment, the files will be copied automatically to the Managed Servers to which the application is targeted.
    - Make the application accessible from the source location that it will be deployed on. You must ensure that each target can reach the location.
13. In Other Options, you can also select one of the following for Deployment Plan Source Accessibility:
    - Use the same accessibility as the application.
    - Copy the deployment plan onto every target. During deployment, the files will be copied automatically to the Managed Servers to which the application is targeted.
    - Make the deployment plan accessible from the source location that it will be deployed on. You must ensure that each target can reach the location.
14. Click **Next.**

    The Deploy Jakarta EE Application: 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.

See About the Common Deployment Tasks in Fusion Middleware Control for more detailed information about these tasks.

Depending on the type of application, in the Deployment Tasks section, you can:

- Configure Web modules: Click **Go to Task** in the Configure Web Modules row. The Configure Web Modules page is displayed. Click **Configure General Properties** to view and edit the general configuration for the Web Module or **Map Resource References** to map the resource references.

  For example, you can change the session invalidation interval or the maximum age of session cookies.

- Configure EJB modules: Click **Go to Task** in the Configure EJB modules row to set standard EJB deployment descriptor properties. The Configure EJB Modules page is displayed. Click **Configure EJB Properties** to view and edit the general configuration for the EJBs or **Map Resource References** to map the resource preferences.

  For example, you can configure the maximum number of beans in the free pool or the network access point.

- Configure application security: Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed, as described in About the Common Deployment Tasks in Fusion Middleware Control.

- Configure persistence: Click **Go to Task** in the Configure Persistence row to configure Java Persistent API (JPA) persistence units.

16. 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.

17. Click **Deploy.**

    Fusion Middleware Control displays processing messages.

18. When the deployment is completed, click **Close.**

Deploying Jakarta EE Applications Using WLST

You can deploy an application using the WLST command line. To deploy a Jakarta EE application when WLST is connected to the Administration Server, you use the WLST command `deploy`, using the following format:

```
deploy(app_name, path [,targets] [,stageMode] [,planPath] [,options])
```

You must invoke the deploy command on the computer that hosts the Administration Server.

For example, to deploy the application mainWebApp:

```
deploy("myApp","/scratch/applications/wlserveR/samples/server/examples/build/mainWebApp")
```

You can also deploy the application (in non-secure mode) using the weblogic.deployer, as shown in the following example:

```
java weblogic.Deployer -adminurl http://localhost:7001
   -user username -password password -deploy
   -name myApp c:\localfiles\mainWebApp
   -plan c:\localfiles\productionEnvPlan.xml
```

For more information about using WLST to deploy applications, see:

- Deployment Tools in *Deploying Applications to Oracle WebLogic Server* for more information about using WLST to deploy applications
- *WLST Command Reference for Oracle WebLogic Server*

Undeploying Jakarta EE Applications

You can undeploy an application or a specific version of an application from a Managed Server instance or a cluster. This section describes how to undeploy an application from a Managed Server. If an application has been deployed to multiple servers, when you undeploy it using Fusion Middleware Control, the application is undeployed from all the servers.

This section contains the following topics:

Undeploying Jakarta EE Applications Using Fusion Middleware Control

To undeploy a Jakarta 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 Deployment menu, choose **Undeploy.**
4. Select the application.
5. In Confirmation page, click **Undeploy.**

   Processing messages are displayed.
6. When the operation completes, click **Close.**

Alternatively, you can navigate to the domain, Managed Server, or cluster. Then, from the target's menu, choose **Application Deployment,** then **Undeploy.** In the Select Application page, select the application you want to undeploy.

Undeploying Jakarta EE Applications Using WLST

You can undeploy an application using the WLST command line. To undeploy a Jakarta 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)
```

Redeploying Jakarta 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.

If you are redeploying a non-versioned application or a versioned application with the same version, note the following:

- The file name and path for the archive you are redeploying must be identical to the file name and path you used when you initially deployed the application.

  For example, if the file name and path of the original application was /dua0/staging/myApp.ear, then the revised application must be /dua0/staging/myApp.ear.
- If you initially deployed the application using the Oracle WebLogic Remote Console or WLST or other management tools other than Fusion Middleware Control, then you cannot redeploy the application using Fusion Middleware Control.

  The following topics describe how to redeploy an application to a Managed Server:
- Redeploying Jakarta EE Applications Using Fusion Middleware Control
- Redeploying Jakarta EE Applications Using WLST

Redeploying Jakarta EE Applications Using Fusion Middleware Control

To redeploy a Jakarta EE application to a Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments.**
2. Select the application to redeploy.

   The application home page is displayed.
3. From the Domain Application Deployment menu, choose **Deployments.**
4. From the Deployments menu, select **Redeploy** to open the Redeploy Jakarta EE Application Assistant.

   The Select Application page is displayed.
5. Click **Next.**
6. In the Archive or Exploded Directory section, you can select one of the following:
   - **Use the archive or exploded directory in the existing source location of the application on the Administration Server.**
   - **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.**
   - **Use the current deployment plan of this application.**

- **Deployment plan is on the machine where this Web browser is running.** Enter the path to the plan or click **Browse** to find the plan file.
- **Deployment plan is on the server where Enterprise Manager is running.** Enter the path to the plan or click **Browse** to find the plan file.

8. Click **Next.**

   The Application Attributes page is displayed.

9. Click **Next.**

   The Deployment Wizard, Deployment Settings page is displayed.

10. On this page, you can perform common tasks before deploying your application or you can edit the deployment plan or save it to a disk. Depending on the type of application, in the Deployment Tasks section, you can:

    - Configure Web modules
    - Configure application security
    - Configure EJB modules
    - Configure persistence

    See About the Common Deployment Tasks in Fusion Middleware Control for detailed information about these tasks.

11. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose. If you edit the deployment plan and change descriptor values, those changes are saved to the deployment plan. In addition, the following configurations are saved to the deployment plan:

    - Application attributes
    - Web module configuration
    - EJB configuration

    Application attributes related to MDS are stored in the file adf-config.xml. Application security attributes are stored in weblogic-application.xml.

    Fusion Middleware Control updates the relevant files and repackages the .ear file.

12. Click **Redeploy.**

    Processing messages are displayed.

13. When the operation completes, click **Close.**

To redeploy an application to a cluster, select the cluster. Then, from the target's menu, select **Application Deployment,** then **Redeploy.**

Redeploying Jakarta EE Applications Using WLST

You can redeploy an application using the WLST command line. To redeploy a Jakarta 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')
```

Deploying, Undeploying, and Redeploying Oracle ADF Applications

Oracle ADF is an end-to-end application framework that builds on Java Platform, Enterprise Edition (Jakarta EE) standards and open-source technologies to simplify and accelerate implementing service-oriented applications.

You can use Fusion Middleware Control, Oracle WebLogic Remote Console, Oracle JDeveloper, or the command line to deploy, undeploy, or redeploy an Oracle ADF application.

See *Developing Fusion Web Applications with Oracle Application Development Framework* for information on developing ADF applications and for deploying them using Oracle JDeveloper

The following topics describe using Fusion Middleware Control, the Remote Console, and the command line to accomplish these tasks:

- Deploying Oracle ADF Applications
- Undeploying Oracle ADF Applications
- Redeploying Oracle ADF Applications

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 and assumes that you have created an .ear file containing the ADF application.

This section contains the following topics:

- Deploying ADF Applications Using Fusion Middleware Control
- Deploying ADF Applications Using WLST

Deploying ADF Applications Using Fusion Middleware Control

To deploy an Oracle ADF application using Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the server in which you want to deploy the application.

   The server home page is displayed.
3. From the WebLogic Server menu, choose **Deployments.**

   The Deployments page is displayed.
4. From the Deployments menu, select **Deploy** to open the Deploy Jakarta EE Application Assistant.

   The Select Archive page is displayed.
5. In the Archive or Exploded Directory section, you can select one of the following:
   - **Archive is on the machine where this browser is running.** Enter the location of the archive or click **Browse** to find the archive file.
   - **Archive or exploded directory is on the server where Enterprise Manager is running.** Enter the location of the archive or click **Browse** to find the archive file.

6. In the Deployment Plan section, you can select one of the following:
   - **Create a new deployment plan when deployment configuration is done.**
   - **Deployment plan is on the machine where this Web browser is running.** Enter the path to the plan.
   - **Deployment plan is on the server where Enterprise Manager is running.** Enter the path to the plan.
7. In the Deployment Type section, you can select one of the following:
   - **Deploy this archive or exploded directory as an application**
   - **Deploy this archive or exploded directory as a library**
8. Click **Next.**

   The Select Target page is displayed.
9. Select the target to which you want to deploy the application. 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 Remote Console.
10. Click **Next.**

    The Application Attributes page is displayed.
11. In the Application Attributes section, for **Application Name,** enter the application name.
12. In the Context Root of Web Modules section, if the Web module does not have the context root configured in the application.xml file, you can specify the context root for your application. The context root is the URI for the Web module. Each Web module or EJB module that contains Web services may have a context root.
13. In the Target Metadata Repository section, you can choose the repository and partition for this application. If the partition name is not specified in the adf-config.xml file, the application name plus the version is used as the default partition name. This ensures that the partition used is unique in the domain so that the metadata for different applications are not accidentally imported into the same repository partition and overwrite each other. Typically, each application's metadata is deployed to its own partition.
    - To change the repository, click the icon next to the **Repository Name.** In the Metadata Repositories dialog box, select the repository and click **OK.**
    - To change the partition, enter the partition name in **Partition Name.** Oracle recommends that you create a new partition for each application. If you enter a name of a partition that does not exist, the partition is created.

    The adf-config.xml file in the .ear file is updated with the new information.

    If the partition or repository specified in the application is not valid in the domain, Fusion Middleware Control displays a message.
14. If the application's adf-config.xml file archive contains MDS configuration for an MDS shared repository, the Shared Metadata Repository section is displayed. In this section, you can choose the repository and partition for this application. If the partition or repository specified in the application is not valid in the domain, Fusion Middleware Control displays a message.

    If you change the repository or partition, the adf-config.xml file in the .ear file is updated with the new information.
15. In the Distribution section, you can select one of the following:

- **Install and start application (servicing all requests)**
- **Install and start application in administration mode (servicing only administration requests)**
- **Install only. Do not start.**

16. You can expand Other Options. See Deploying Jakarta EE Applications for a description of those options.
17. Click **Next.**

    The Deployment Wizard, Deployment Settings page is displayed.
18. On this page, you can perform common tasks before deploying your application or you can edit the deployment plan or save it to a disk. Depending on the type of application, in the Deployment Tasks section, you can:

    - Configure Web modules: Click **Go to Task** in the Configure Web Modules row. The Configure Web Modules page is displayed. Click **Configure General Properties** to view and edit the general configuration for the Web Module or **Map Resource References** to map the resource references.

      For example, you can change the session invalidation interval or the maximum age of session cookies.
    - Configure EJB modules: Click **Go to Task** in the Configure EJB modules row to set standard EJB deployment descriptor properties. The Configure EJB Modules page is displayed. Click **Configure EJB Properties** to view and edit the general configuration for the EJBs or **Map Resource References** to map the resource preferences.

      For example, you can configure the maximum number of beans in the free pool or the network access point.
    - Configure application security: Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed, as described in About the Common Deployment Tasks in Fusion Middleware Control.
    - Configure persistence: Click **Go to Task** in the Configure Persistence row to configure Java Persistent API (JPA) persistence units.
    - Configure ADF Connections: To modify the ADF connections, click **Go to Task** in the Configure ADF Connections row. The Configure ADF Connections page is displayed, showing the current connection information. To modify a connection type, click the **Edit** icon for a particular row. For example, you can modify the connection information for an external application. For more information about ADF connections, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

    For more information about these options, see About the Common Deployment Tasks in Fusion Middleware Control.
19. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.
20. Click **Deploy.**

    Fusion Middleware Control displays processing messages.
21. When the deployment is completed, click **Close.**

Deploying ADF Applications Using WLST

To deploy an ADF application using the WLST command line:

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:

   - Deployment Tools in *Deploying Applications to Oracle WebLogic Server* for more information about using WLST to deploy applications
   - *WLST Command Reference for Oracle WebLogic Server*

Undeploying Oracle ADF Applications

To undeploy an Oracle ADF application using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments.**
2. Select the application to undeploy.

   The application home page is displayed.
3. From the Domain Application Deployment menu, choose **Deployments.**
4. From the Deployments menu, select **Undeploy.**
5. In the Confirmation page, click **Undeploy.**

   Processing messages are displayed.
6. When the operation completes, click **Close.**

Alternatively, you can navigate to the domain, Managed Server, or cluster. Then, from the target's menu, choose **Application Deployment,** then **Undeploy.** In the Select Application page, select the application you want to undeploy.

Note that when you undeploy an application, documents stored in the MDS partition are not deleted.

Redeploying Oracle ADF Applications

When you redeploy an application, if the application contains a Metadata Archive (MAR), the contents of the MAR is imported to the application's metadata repository only if the MAR is changed. If the MAR is unchanged from previous deployment of the application, it is ignored.

If you are redeploying a non-versioned application or a versioned application with the same version, note the following:

- The file name and path for the archive you are redeploying must be identical to the file name and path you used when you initially deployed the application.

  For example, if the file name and path of the original application was /dua0/staging/myApp.ear, then the revised application must be /dua0/staging/myApp.ear.
- If you initially deployed the application using the Oracle WebLogic Remote Console or WLST or other management tools other than Fusion Middleware Control, then you cannot redeploy the application using Fusion Middleware Control.

To redeploy an Oracle ADF application using Fusion Middleware Control:

1. From the navigation pane, expand **Application Deployments.**
2. Select the application to redeploy.

   The application home page is displayed.
3. From the Domain Application Deployment menu, choose **Deployments.**
4. From the Deployments menu, select **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:
   - **Use the archive or exploded directory in the existing source location of the application on the Administration Server.**
   - **Archive is on the machine where this browser is running.** Enter the location of the archive or click **Browse** to find the archive file.
   - **Archive or exploded directory is on the server where Enterprise Manager is running.** Enter the location of the archive or click **Browse** to find the archive file.
7. In the Deployment Plan section, you can select one of the following:
   - **Create a new deployment plan when deployment configuration is done.**
   - **Deployment plan is on the machine where this web browser is running.** Enter the path to the plan.
   - **Deployment plan is on the server where Enterprise Manager is running.** Enter the path to the plan.
8. Click **Next.**

   The Application Attributes page is displayed.
9. In the Application Attributes section, for **Application Name,** enter the application name.
10. In the Context Root of Web Modules section, if the Web module does not have the context root configured in the application.xml file, you can specify the context root for your application. The context root is the URI for the Web module. Each Web module or EJB module that contains Web services may have a context root.

11. The Target Metadata Repository section is displayed. In this section, you can choose the repository and partition for this application:
    - To change the repository, click the icon next to the **Repository Name.** In the Metadata Repositories dialog box, select the repository and click **OK.**
    - To change the partition, enter the partition name in **Partition Name.** Oracle recommends that you create a new partition for each application. If you enter a name of a partition that does not exist, the partition is created.
12. If the application's adf-config.xml file archive contains MDS configuration for an MDS shared repository, the Shared Metadata Repository section is displayed. In this section, you can choose the repository and partition for this application.
13. Click **Next.**

    The Deployment Settings page is displayed.
14. On this page, you can perform common tasks before deploying your application or you can edit the deployment plan or save it to a disk. In the Deployment Tasks section, you can:
    - Configure Web modules
    - Configure application security
    - Configure persistence

    See About the Common Deployment Tasks in Fusion Middleware Control for detailed information about these options.
15. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.
16. Click **Deploy.**

    Fusion Middleware Control displays processing messages.
17. When the deployment is completed, click **Close.**
18. In the Confirmation page, click **Redeploy.**

Deploying, Undeploying, and Redeploying SOA Composite Applications

Oracle SOA Suite uses the SCA standard as a way to assemble service components into a SOA composite application. You can deploy, undeploy, and redeploy 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 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.

For additional information about deploying SOA composite applications, see *Administering Oracle SOA Suite and Oracle Business Process Management Suite*

The following topics describe using Fusion Middleware Control to accomplish these tasks:

- Deploying SOA Composite Applications
- Undeploying SOA Composite Applications
- Redeploying SOA Composite Applications

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 **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.

3. 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.

     You can 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 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. In the SOA Partition section, select the partition into which to deploy this SOA composite application. Partitions enable you to logically group SOA composite applications into separate sections. Note that even if there is only one partition available, you must explicitly select it. Once deployed, a composite cannot be transferred to a different partition.

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, close the confirmation box.

See Deploying Applications in *Administering Oracle SOA Suite and Oracle Business Process Management Suite* for complete information about deploying SOA Composite applications.

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 **SOA,** and then select **soa-infra.**

   The SOA Infrastructure home page is displayed.

2. From the SOA Infrastructure menu, choose **SOA Deployment,** then **Undeploy.**
3. Select the composite to undeploy and click **Next.**
4. Review your selections. If you are satisfied, click **Undeploy.**

   Processing messages are displayed.

5. When undeployment has completed, close the confirmation window.

See Undeploying Applications in *Administering Oracle SOA Suite and Oracle Business Process Management Suite* for complete information about undeploying SOA Composite applications

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 **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. Click **Next.**

   The Select Archive page appears.

5. In the Archive or Exploded Directory section, select the location of the SOA composite application revision you want to redeploy.
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 Redeploying Applications in *Administering Oracle SOA Suite and Oracle Business Process Management Suite* for complete information about redeploying SOA Composite applications

Deploying, Undeploying, and Redeploying WebCenter Portal Applications

Oracle WebCenter Portal applications differ from traditional Jakarta 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 Portal application customizations are stored in Oracle Metadata Services (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 Portal application.

For more information about deploying WebCenter Portal applications, see *Administering Oracle WebCenter Portal*

The following topics describe using Fusion Middleware Control to accomplish these tasks:

- Deploying WebCenter Portal Applications
- Undeploying WebCenter Portal Applications
- Redeploying WebCenter Portal Applications

Deploying WebCenter Portal Applications

To deploy your application to a Managed Server that resides outside JDeveloper, you must first create an application deployment plan. 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 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 *Developing for Oracle WebCenter Portal*.

For complete information about deploying Oracle WebCenter Portal applications, see Deploying WebCenter Portal Framework Applications in *Administering Oracle WebCenter Portal*.

To deploy an Oracle WebCenter Portal application to a Managed Server using Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the domain in which you want to deploy the application.

   The server home page is displayed.
3. From the WebLogic Domain menu, select **Deployment.** then **Deploy.**

4. From the Deployments menu, select **Deploy** to open the Deploy Jakarta EE Application Assistant.

   The Deployment Wizard, 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 Select Target page is displayed.

8. 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.

9. Click **Next.**

   The Application Attributes page is displayed.

10. In the Application Attributes section, for **Application Name,** enter the application name.

11. 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.

12. 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.

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, in the Deployment Tasks section, you can:

    - Configure Web modules: Click **Go to Task** in the Configure Web Modules row. The Configure Web Modules page is displayed. Click **Configure General Properties** to view and edit the general configuration for the Web Module or **Map Resource References** to map the resource references.

For example, you can change the session invalidation interval or the maximum age of session cookies.

- Configure EJB modules: Click **Go to Task** in the Configure EJB modules row to set standard EJB deployment descriptor properties. The Configure EJB Modules page is displayed. Click **Configure EJB Properties** to view and edit the general configuration for the EJBs or **Map Resource References** to map the resource preferences.

  For example, you can configure the maximum number of beans in the free pool or the network access point.

- Configure application security: Click **Go to Task** in the Configure Application Security row. Depending on what type of security is used, different pages are displayed, as described in About the Common Deployment Tasks in Fusion Middleware Control.
- Configure persistence: Click **Go to Task** in the Configure Persistence row to configure Java Persistent API (JPA) persistence units.
- Configure ADF Connections: To modify the ADF connections, click **Go to Task** in the Configure ADF Connections row. The Configure ADF Connections page is displayed, showing the current connection information. To modify a connection type, click the **Edit** icon for a particular row.

See About the Common Deployment Tasks in Fusion Middleware Control for more detailed information about these options.

15. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.

16. Click **Deploy.**

    Fusion Middleware Control displays processing messages.

17. When the deployment is completed, click **Close.**

Undeploying WebCenter Portal Applications

To undeploy a WebCenter Portal application:

1. From the navigation pane, expand **Application Deployments,** then select the application to undeploy.

   The application home page is displayed.

2. From the Deployments menu, select **Undeploy** to open the Deploy Jakarta EE Application Assistant.

   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.

Redeploying WebCenter Portal Applications

To redeploy a WebCenter Portal application:

1. From the navigation pane, expand the domain.

2. Select the server in which you want to redeploy the application.

   The server home page is displayed.

3. From the WebLogic Server menu, select **Deployments.**

   The Deployments 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. From the Deployments menu, select **Redeploy** to open the Deploy Jakarta EE Application Assistant.

6. Click **Next.**

   The Redeploy Application page is displayed.

7. Click **Next.**

   The Select Archive page is displayed.

8. 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.

9. In the Deployment Plan section, you can select one of the following:

   - **Create a new deployment plan when deployment configuration is done**
   - **Use the current deployment plan of this application.**
   - **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.

10. Click **Next.**

    The Application Attributes page is displayed.

11. In the Application Attributes section, for **Application Name,** enter the application name.

12. In the Context Root of Web Modules section, 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.

13. In the Target Metadata Repository section, select the MDS Repository and for **Partition Name,** enter a partition name. Be careful to use the same repository connection and partition name that you used when you originally deployed the application. If you do not, all customizations are lost.

14. Click **Next.**

    The 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. In the Deployment Tasks section, you can:

    - Configure Web modules
    - Configure application security
    - Configure persistence

See About the Common Deployment Tasks in Fusion Middleware Control for detailed information about these options.

16. Expand **Deployment Plan.**

    You can edit and save the deployment plan, if you choose.

17. Click **Redeploy.**

    Fusion Middleware Control displays processing messages.

18. When the deployment is completed, click **Close.**

Managing Deployment Plans

A **deployment plan** is a client-side aggregation of all the configuration data needed to deploy an archive into Oracle WebLogic Server. A deployment plan allows you to easily deploy or redeploy an application using a saved set of configuration settings.

A new deployment plan is created by default if you do not apply an existing deployment plan to an application at the time of deployment, as described in Deploying Jakarta EE Applications. Once created, you can save a deployment plan as a file and reuse it for redeploying the application or for deploying other applications.

However, if you change the configuration of an application after it is deployed (for example, if you modify the MDS configuration of an application), then any existing deployment plans you saved no longer represent the configuration settings of the deployed application.

In such a situation, you can fetch a new deployment plan that more closely represents the configuration of the deployed application.

To fetch the deployment plan of an application that is currently deployed:

1. From the **WebLogic Domain** menu, choose **Deployments.**
2. Select an application from the list of currently deployed applications.
3. From the Deployments menu, select **Fetch Deployment Plan.**

   The Fetch Deployment Plan page is displayed.

4. Select a location where you want to save the deployment plan, and click **Fetch**.

   You can save the plan to the computer where the Web browser is running or to the computer where Fusion Middleware Control is running.

5. In the resulting dialog box, specify a directory location for the saved deployment plan.

   You can now use this deployment plan to later deploy or redeploy the application using the configuration currently in use by the application.

Alternatively, you can edit a deployment plan on the Deployment Settings page of the Application Deployment wizard.

About the Common Deployment Tasks in Fusion Middleware Control

When you deploy an application using Fusion Middleware Control, you can use the Deployment Settings page of the Deployment wizard to perform specific deployment configuration tasks before the application is deployed.

The following describes the deployment tasks that can appear on the Deployment Settings page, depending on the type of application you are deploying.

Configure Web modules

This deployment task is available when you are deploying any application that includes a Web module. In most cases, this means the application contains a Web application deployment descriptor (web.xml or weblogic.xml); however, a Web module can also be identified by annotations in the Java code of the application.

You can use this deployment task to set standard Web application deployment descriptor properties, such as:

- Session validation interval
- Maximum age of session cookies

Configure EJBs

This deployment task is available for any application that includes an EJB module. In most cases, this means the application contains an EJB deployment descriptor (ejb-jar.xml or weblogic-ejb-jar.xml); however, an EJB module can also be identified by annotations in the Java code of the application.

You can use this deployment task to set standard EJB deployment descriptor properties, such as:

- The maximum number of beans in the free pool
- The EJB network access point

Configure Application Security

This deployment task is available for all application types. However, the options available when you select this task vary depending on the existence of the following files in the application:

- jazn-data.xml

  If the jazn-data.xml file exists in the application, then you can:

  - Append, overwrite, or ignore policy migration.
    - If you are deploying the application for the first time, then select **Append**.
    - If the application was previously deployed and the application authorization policy exists, then select **Append**, or select **Ignore** to keep the application authorization policy.
    - To overwrite the previous policy, then select **Overwrite**.
  - Specify the Application stripe ID, if the stripe ID is inconsistent with the one defined in the migration options.
  - Specify that policies are removed when the application is undeployed.

- cwallet.sso

  If an cwallet.sso file exists in the application, then you can set additional application credential migration options.

If the application contains both files, the page displays both sections.

For more information about the settings available when you select the Configure Application Security deployment task, see Deploying Java EE and Oracle ADF Applications with Fusion Middleware Control in *Securing Applications with Oracle Platform Security Services*.

If neither of these files exists in the application, then you can use this task to determine how user roles and policies will be defined when the application is deployed. For example, you can choose to use only the roles and policies defined in the deployment descriptors, or you can choose to use only the roles and policies defined on the server. The Configure Application Security page displays the following options:

- **Deployment Descriptors Only:** Use only roles and policies that are defined in the deployment descriptors.
- **Custom Roles:** Use roles that are defined in the Remote Console; use policies that are defined in the deployment descriptor.
- **Custom Roles and Policies:** Use only roles and policies that are defined in the Remote Console.
- **Advanced:** Use a custom model that you have configured on the realm's configuration page.

Configure persistence

This deployment task is available for applications that contain one or more persistence.xml files. Using this task, you can configure the Java Persistent API (JPA) persistence units for the application.

You can view details about each persistence unit and define a Java Transaction API (JTA) data source or non-JTA data source for each persistence unit.

Configuring the data sources for persistence units can be useful for applications that take advantage of Oracle TopLink. See the Solutions Guide for Oracle TopLink.

For more information about how persistence units and the persistence.xml file can be used in Jakarta EE applications, refer to the definition of Persistence Units in the Jakarta EE 5 Tutorial at the following Web site:

http://download.oracle.com/javaee/5/tutorial/doc/bnbqw.html#bnbrj

Configure ADF connections

This deployment task is available for applications that use ADF connections. You can modify the connection information for an external application. For more information about ADF connections, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

Configuring Applications in Fusion Middleware Control

With Fusion Middleware Control, you can configure attributes of your application.

The type of attributes that you can configure depends on the type of application. For example, for enterprise applications, you can configure deployment order, the maximum age of session cookies, the Managed Servers or clusters to which you want to deploy the application, and other attributes.

1. From the WebLogic Domain menu, select **Deployments.**
2. Select an application.

   The application’s summary page is displayed.
3. From the context menu, such as Domain Application Deployment, select an option, such as Administration.

   The configuration attributes that you can modify are displayed.

4. If the application does not have a deployment plan, click **Create New Deployment Plan.** Specify a location for the plan.
5. Modify any attributes you want to change or provide values for attributes with none.
6. Click **Apply.**

Changing MDS Configuration Attributes for Deployed Applications

If your application uses an MDS Repository, you can modify configuration attributes after the application is deployed.

To view or modify the attributes, you can use the System MBean Browser or WLST.

> **Note:**
>
> Changes to the configuration persist in MDS as customizations. Because these persist as customizations:
>
> - Any changes made to the configuration are retained across application deployments. For example, assume that an application has an ExternalChangeDetectionInterval configuration attribute value set to 40 seconds through Oracle JDeveloper. If you change the ExternalChangeDetectionInterval configuration attribute to 50 seconds, and you redeploy the application, the value of the attribute remains at 50 seconds.
> - In a cluster, because all instances of the deployed application point to the same MDS Repository partition, all instances of the application use the same value. If a configuration attribute has been changed for one application instance, all instances of that application in a cluster use the changed value.

The following topics describe how you can change the MDS configuration attributes:

- Changing the MDS Configuration Attributes Using Fusion Middleware Control
- Changing the MDS Configuration Using WLST
- Restoring the Original MDS Configuration for an Application

Changing the MDS Configuration Attributes Using Fusion Middleware Control

To change the MDS configuration attributes of an application, take the following steps:

1. Navigate to the application's home page by expanding **Application Deployments.** Then, select an application.

   The application's home page is displayed.
2. From the Application Deployment menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.
3. Expand **Application Defined MBeans,** then **oracle.adf.share.config,** then **Server: *name***, then **Application: *name***, then **ADFConfig,** then **ADFConfig,** and **ADFConfig.**

4. Select **MDSAppConfig.**

   The Application Defined MBeans page is displayed.

5. You can view the description and values for the attributes.

   Table 10-2 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 10-2 MDS Configuration Attributes for Deployed Applications

Attribute	Description
AppMetadataRepositoryInfo	Read only. Describes the metadata repository partition where the application is deployed.
AutoPurgeTimeToLive	Automatically purge versions of metadata documents older than the given time interval, specified in seconds. Any unlabeled versions older than this time interval are automatically purged on any subsequent update from this application. If the value is not set, versions are not automatically purged.
ConfigMBean	If true, indicates that this MBean is a Config MBean.
DeployTargetRepository	The name of the target repository configured for the application.
eventProvider	If true, it indicates that this MBean is an event provider as defined by the Jakarta Management Specification.
eventTypes	All the event's types emitted by this MBean.
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, all of 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.

Table 10-2 (Cont.) MDS Configuration Attributes for Deployed Applications

Attribute	Description
MaximumCacheSize	The maximum metadata cache size limit, in kilobytes. If the value is 0, caching is disabled. If no value is specified, there is no cache limit. In this case, cached data is stored indefinitely.
objectName	All the event's types emitted by this MBean.
ReadOnly	If true, it indicates that this MBean is a read-only MBean.
ReadOnlyMode	Changes the application to read-only mode, so that no updates can be made to the application's repository partition, including configuration and application metadata.
RemoteNotifications	Enables distributed remote notifications of applicable metadata changes.This parameter is valid only if ExternalChangeDetection is enabled.
RestartNeeded	Enables distributed remote notifications of applicable metadata changes.This parameter is only valid if ExternalChangeDetection is enabled.
RetryConnection	Enables the application to retry the connection to the metadata repository after connection failure.
SharedMetadataRepositoryInfo	Read only. Specifies the MDS Repository partition used by the application. Note that an application can use more than one shared metadata repository.
stateManageable	If true, it indicates that this MBean provides State Management capabilities as defined by the Jakarta Management Specification.
statisticsProvider	If true, it indicates that this MBean is a statistic provider as defined by the Jakarta Management Specification.
SystemMBean	If true, it indicates that this MBean is a System MBean.
Visible	If true, it indicates that this MBean is visible to the current user.

6. To view or modify an attribute, select the attribute.

   The attribute page is displayed.

7. If the attribute is not read-only, you can change the values. For example, for AutoPurgeTimeToLive, you can change the interval, by entering a new value in **Value.**
8. Click **Apply.**
9. Navigate up to ADFConfig (the parent of MDSAppConfig) and select it.
10. In the Operations tab, click **Save.**
11. Click **Invoke.**

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)
```

Restoring the Original MDS Configuration for an Application

To restore the original MDS configuration for an application:

1. Navigate to the application's home page by expanding **Application Deployments.** Then, select an application.

   The application's home page is displayed.

2. From the Application Deployment menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.

3. Expand **Application Defined MBeans,** then **oracle.adf.share.config,** then **Server: *name***, then **Application: *name***, then **ADFConfig,** and then **ADFConfig.**

4. Select the Operations tab.

5. Select **RestoreToOriginalConfiguration.**

   The Operation: restoreToOriginalConfiguration page is displayed.

6. Click **Invoke.**

   Use this operation with caution. It causes all changes made to the original adf-config.xml file to be discarded. The adf-config.xml is restored to the base document.

Part V
Monitoring Oracle Fusion Middleware

It is important to know how to monitor Oracle Fusion Middleware, how to view and manage log files to assist in monitoring system activity, find information about the cause of an error and its corrective action, and to diagnose problems.

11

Monitoring Oracle Fusion Middleware

You can monitor Oracle Fusion Middleware using Fusion Middleware Control, Oracle WebLogic Remote Console, and the command line.

- 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.
- 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.

Monitoring the Status of Oracle Fusion Middleware

Monitoring the health of your Oracle Fusion Middleware environment and ensuring that it performs optimally is an important task for the administrator.

Oracle Fusion Middleware provides the following methods for monitoring the status of your environment:

- Fusion Middleware Control: You can monitor the status of Oracle WebLogic Server domains, clusters, servers, Java components, system components, and applications. Navigate to the entity's home page, for example, to the home page for an Oracle HTTP Server instance.
- Oracle WebLogic Remote Console: You can monitor the status of Oracle WebLogic Server domains, clusters, servers, Java components, and applications. From the Administration Console, navigate to the entity's page.
- The command line: You can monitor the status of your environment using the WLST state command.

Most of the monitoring tasks in this chapter describe how to monitor using Fusion Middleware Control or the command line.

The following topics provide more detail:

- Monitoring an Oracle WebLogic Server Domain
- Monitoring an Oracle WebLogic Server Administration or Managed Server
- Monitoring a Cluster
- Monitoring a Java Component
- Monitoring a System Component
- Monitoring Jakarta EE Applications
- Monitoring ADF Applications
- Monitoring the SOA Infrastructure and SOA Composite Applications
- Monitoring Oracle WebCenter Portal Applications
- Monitoring Applications Deployed to a Cluster

- Monitoring the Status of Components Using the Command Line

Monitoring an Oracle WebLogic Server Domain

You can view the status of a domain, including the servers, clusters, and deployments in the domain from the domain home page of Fusion Middleware Control:

1. From the WebLogic Domain menu, select **Home.**

   The domain home page is displayed. This page shows the following:

   - The name of the Administration Server and the host on which it is located.
   - A table listing information about the servers in the domain. The table contains the columns Name, Status, Server Type, Host, Cluster, Listen Port, CPU Usage, and Heap Usage.

2. For more specific monitoring information about the domain, from the WebLogic domain menu, select Monitoring, then other subcategories, such as performance summary, health, or deployments.

Monitoring an Oracle WebLogic Server Administration or Managed Server

You can view the status of a WebLogic Server Administration Server or Managed Server in Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the server.

   The server home page is displayed. 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

3. For more specific monitoring information about the server, from the WebLogic Server menu, select Monitoring, then other subcategories, such as performance summary, health, or deployments.

Monitoring a Cluster

You can view the status of a cluster, including the servers and deployments in the cluster using Fusion Middleware Control:

1. From the navigation pane, expand the domain.
2. Select the cluster.

   The **Cluster** page shows general information about the cluster, including the cluster messaging mode, response and load and information about servlets and JSPs.

3. To see a summary of the status of the servers in the cluster, rom the WebLogic Cluster menu, select Monitoring, then Summary.
4. For more specific monitoring information about the cluster, from the WebLogic Cluster menu, select Monitoring, then other subcategories, such as Performance Summary, Health, or Deployments.

Monitoring a Java Component

You can view the status of a Java component, including whether the component is started, in the component home page in Fusion Middleware Control.

To monitor a Java component, such as Oracle SOA SuiteOracle Enterprise Scheduler:

1. From the navigation pane, expand the type of component, such as SOA, then soa-infra.
2. Select the component. For example, select soa-infra.

   The component home page is displayed. This page shows general information about the component, including its state and key metrics. The information shown depends on the type of component.
3. For more specific monitoring information about the component, from the component menu, select Monitoring, then performance summary.

Monitoring a System Component

To monitor a system component, such as Oracle HTTP Server:

1. From the navigation pane, expand the component type, such as **HTTP Server.**
2. Select the component, such as ohs1.

The component home page is displayed. This page shows the following:

- A General section with basic information about the component, such as name and state
- A response and load section, which shows the requests per second and the request processing time
- CPU and memory usage

Monitoring Jakarta EE Applications

To monitor a Jakarta 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 applications home page is displayed. This page shows a summary of the application, including its state.

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.

   You can view the following information in the application’s home page:

- A summary of the application, including its state, the Managed Server on which it is deployed, and information about active sessions, active requests, and request processing time
- Deployments, which lists the servers on which the application is deployed
- A list of modules with the type of module for each, EJB Components, and web services
- A list of data sources

2. To view health of the environment, from the **Application Deployments** menu, choose **Monitoring,** then **Environment Monitoring.** The Environment Monitoring page is displayed.

   It contains tabs for Health, Query Caching, Workload, and Coherence.

   For more information about monitoring ADF applications, see Monitoring and Configuring ADF Applications Using Fusion Middleware Control in *Administering Oracle ADF Applications*.

Monitoring the SOA Infrastructure and SOA Composite Applications

To monitor the SOA Infrastructure and SOA composite applications, see the following:

- Monitoring the Overall Status of the SOA Infrastructure or Individual Partition in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*
- Monitoring SOA Composite Applications in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*

Monitoring Oracle WebCenter Portal Applications

To monitor Oracle WebCenter Portal applications, see Monitoring WebCenter Portal in *Administering Oracle WebCenter Portal*.

Monitoring Applications Deployed to a Cluster

If you deploy an application to a cluster, Oracle Fusion Middleware automatically deploys the application to each Managed Server in the cluster. As a result, there is an instance of the application on each server.

There are times when you want to monitor the performance of the application on an individual server, and times when you want to monitor the overall performance of the application across all the servers in the cluster.

For example, normally, you would manage the overall performance of the application to determine if there are any performance issues affecting all users of the application, regardless of which instance users access. If you notice a performance problem, you can then drill down to a specific instance of the application to determine if the problem is affecting one or all of the application instances in the cluster.

Fusion Middleware Control provides monitoring pages for both of these scenarios:

1. From the navigation pane, expand **Application Deployments**.

   Fusion Middleware Control lists the applications deployed in the current domain.

2. If an application has been deployed to a cluster, expand the application in the navigation pane. Fusion Middleware Control shows that it is deployed to the cluster to indicate that it represents more than one instance of the application on the cluster.

3. Monitor the overall performance of the application on the cluster by clicking the cluster application, or monitor the performance of the application on a single server by clicking one of the application deployment instances.

Monitoring the Status of Components Using the Command Line

To monitor the status of components using the WLST command line:

- For Java components, use the WLST `state` command, with the following format:

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

  For example, to get the status of the Managed Server server1, use the following command:

  ```
  wls:/mydomain/serverConfig> state('server1','Server')
  Current state of "server1": SUSPENDED
  ```

- To monitor the status of system components, use the WLST `state` command, with the following format:

  Oracle Fusion Middleware provides the following methods for monitoring the status of your environment:

  To monitor the status of system components, use the WLST `state` command, with the following format:

  ```
  state('component_name')]
  ```

  For example, to view the status ohs1, use the following command:

  ```
  state('ohs1')]
  ```

Viewing the Performance of Oracle Fusion Middleware

If you encounter a problem, such as an application that is running slowly or is hanging, you can view more detailed performance information, including performance metrics for a particular target, to find out more information about the problem.

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

Note that Fusion Middleware Control provides real-time data. If you are interested in viewing historical data, consider using Oracle Enterprise Manager Grid Control.

For example, to view the performance of an Oracle WebLogic Server Managed Server:

1. From the navigation pane, expand the domain.
2. Select the server to monitor.

   The Managed Server home page is displayed.
3. From the WebLogic Server menu, choose **Monitoring,** then **Performance Summary.**

   The Performance Summary page is displayed. It shows performance metrics, as well as information about response time and request processing time for applications deployed to the Oracle WebLogic Server.
4. To see additional metrics, click **Show Metric Palette** and expand the metric categories.
5. Select a metric to add it to the Performance Summary.

6. You can compare one server with another by selecting Compare, then With Another Oracle WebLogic Server. To overlay another target, click **Overlay**, and select the target. The target is added to the charts, so that you can view the performance of more than one target at a time, comparing their performance.
7. To customize the time frame shown by the charts, you can:
   - Click **Slider** to display a slider tool that lets you specify that more or less time is shown in the charts. For example, to show the past 10 minutes, instead of the past 15 minutes, slide the left slider control to the right until it displays the last 10 minutes.
   - Select the calendar and clock icon. Then, enter the **Start Time** and **End Time.** If there is no data available for those times, a confirmation message displays, explaining the timeline will be automatically adjusted to the time period for which the data is available.

You can also view the performance of a components, such as Oracle HTTP Server or Oracle SOA Suite. Navigate to the component and select **Monitoring,** then **Performance Summary** from the dynamic target menu.

12

Managing Log Files and Diagnostic Data

Oracle Fusion Middleware components generate log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, and access information on HTTP requests.
This chapter describes how to find information about the cause of an error and its corrective action and to view and manage log files to assist in monitoring system activity and in diagnosing problems.

- Overview of Oracle Fusion Middleware Logging
  By default, Oracle WebLogic Server is configured to use the common log format for HTTP access logs. Most other Oracle Fusion Middleware components write diagnostic log files in the Oracle Diagnostic Logging (ODL) format.
- About 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, written in a specific format.
- Viewing and Searching Log Files
  You can view, list, and search log files across Oracle Fusion Middleware components. You can view and search log files using Fusion Middleware Control or you can download a log file to your local client and view the log files using another tool. You can also list, view, and search log files using the WLST command-line tool.
- Configuring Settings for Log Files
  You can change the location of log files, how often the log files rotate, and the level of information written to them, along with other configuration settings. You can change the log settings of Managed Servers and Java components using Fusion Middleware Control or WLST.
- About 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.
- Configuring Tracing
  Sometimes you need more information to troubleshoot a problem than is usually recorded in the logs. One way to achieve that is to increase the level of messages logged by one or more components, and fine-tune which messages are written to the log files.

Overview of Oracle Fusion Middleware Logging

By default, Oracle WebLogic Server is configured to use the common log format for HTTP access logs. Most other Oracle Fusion Middleware components write diagnostic log files in the Oracle Diagnostic Logging (ODL) format.

The following topics describe HTTP access logging and diagnostic logging.

- About Oracle Fusion Middleware HTTP Access Logging
- About Oracle Fusion Middleware Diagnostic Logging

About Oracle Fusion Middleware HTTP Access Logging

By default, Oracle WebLogic Server is configured to use the common log format for HTTP access logs. Oracle WebLogic Server also supports the extended log format, an emerging standard defined by the draft specification from the World Wide Web Consortium (W3C).

When you install Oracle WebLogic Server with Oracle JRF, it uses the extended log format for HTTP access logs by default. The extended log format allows you to specify the type and order of information recorded about each HTTP communication.

Oracle Fusion Middleware supports the following field identifiers:

- date: The date at which transaction completed. The field has the format YYYY-MM-DD. All dates are specified in GMT.
- time: The time at which transaction completed. The field has the format HH:MM, HH:MM:SS or HH:MM:SS.S where HH is the hour in 24-hour format, MM is minutes, and SS is seconds. All times are specified in GMT.
- cs-method: The request method, for example GET or POST. This field has type <name>, as defined in the W3C specification.
- cs-url: The full requested URI. This field has type <uri>, as defined in the W3C specification.
- ctx-ecid: The Execution Context ID (ECID). The ECID is a globally unique identifier associated with the execution of a particular request.
- ctx-rid: The Relationship ID (RID). The RID distinguishes the work done in one thread on one process, from work done by any other threads on this and other processes on behalf of the same request.
- sc-status: The status code of the response, for example (404) indicating a "File not found" status. This field has type <integer>, as defined in the W3C specification.

For information about the extended log format fields, see:

`http://www.w3.org/TR/WD-logfile.html`

About Oracle Fusion Middleware Diagnostic Logging

Most Oracle Fusion Middleware components write diagnostic log files in the **Oracle Diagnostic Logging** (ODL) format. Log file naming and the format of the contents of log files conforms to an Oracle standard. By default, the diagnostic messages are written in text format.

ODL provides the following benefits:

- The capability to limit the total amount of diagnostic information saved. You can set the level of information saved and you can specify the maximum size of the log file and the log file directory.
- When you reach the specified size, older segment files are removed and newer segment files are saved in chronological fashion.
- Components can remain active, and do not need to be shutdown, when older diagnostic logging files are deleted.

You can view log files using Fusion Middleware Control or the WLST `displayLogs` command, or you can download log files to your local client and view them using another tool (for example, a text editor or another file viewing utility).

> **Note:**
>
> Oracle WebLogic Server does not use the ODL format. For information about the Oracle WebLogic Server log format, see Log Message Format in *Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

About 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, written in a specific format.

The following example shows an ODL format error messages from Oracle HTTP Server:

```
[2017-03-13T12:31:29.0584-07:00] [OHS] [NOTIFICATION:16] [OHS-9999]
[mod_weblogic.c] [host_id: example] [host_addr: nn.nnn.nn.nn] [pid: 12789]
 [tid: 46919953675776] [user: username VirtualHost: main
WebLogic Server Plugin version 12.1.2 <WLSPLUGINS_MAIN_LINUX.X64_130502.1731>
```

In the message, the fields map to the following attributes, which are described in Table 12-1:

- Timestamp, originating: `2017-03-13T12:31:29.0584-07:00`
- Organization ID: `OHS`
- Message Type: `NOTIFICATION:16`
- Component ID: `mod_weblogic.c`
- Host ID: `host_id: example`
- Host Address: `host_addr: nn.nnn.nn.nn`
- Process ID: `pid: 12789`
- Thread ID: `tid: 46919953675776`
- User ID: `userId: username`
- Virtual Host: *`VirtualHost: main`*
- Message Text: `"WebLogic Server Plugin version 12.1.2 <WLSPLUGINS_MAIN_LINUX.X64_130502.1731>"`

By default, the information is written to the log files in ODL text format. You can change the format to ODL XML format, as described in Specifying the Log File Format.

Table 12-1 describes the contents of an ODL message. For any given component, the optional attributes may not be present in the generated diagnostic messages.

Table 12-1 ODL Format Message Fields

Attribute Name	Description	Required
Timestamp, Originating (TIME)	The date and time when the message was generated. This reflects the local time zone.	Yes
Timestamp, normalized (time_norm)	The timestamp normalized for clock drift across hosts. This field is used when the diagnostic message is copied to a repository on a different host.	No
Organization ID (org_id)	The organization ID for the originating component.	No

Table 12-1 (Cont.) ODL Format Message Fields

Attribute Name	Description	Required
INSTANCE_ID (INST_ID)	The name of the instance to which the component that originated the message belongs.	No
COMPONENT ID (COMP_ID)	The ID of the component that originated the message.	Yes
MESSAGE_ID (MSG_ID)	The ID that uniquely identifies the message within the component. The ID consists of a prefix that represents the component, followed by a dash, then a 5-digit number. For example: `OHS-51009`	Yes
MESSAGE_TYPE (MSG_TYPE)	The type of message. Possible values are: INCIDENT_ERROR, ERROR, WARNING, NOTIFICATION, TRACE, and UNKNOWN. See Table 12-3 for information about the message types.	Yes
MESSAGE_LEVEL (MSG_LEVEL)	The message level, represented by an integer value that qualifies the message type. Possible values are from 1 (highest severity) through 32 (lowest severity). See Table 12-3 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 About Correlating Messages Across Log Files and Components 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 About Correlating Messages Across Log Files and Components for information about RIDs.	No
SUPPL_ATTRS	An additional list of name/value pairs which contain component-specific attributes about the event. Oracle Fusion Middleware provides the supplemental attribute DSID (Diagnostic Session ID). DSID is an ID for a user session and is used to map a set of log messages, incidents, and other diagnostic data to a user session. For example, you can see if a specific incident generated in a user session may have been proceeded by earlier incidents in the same session, and could therefore be the root cause of the subsequent incident.	No
MESSAGE TEXT (TEXT)	The text of the message.	Yes
Message Arguments (arg)	A list of arguments bound with the message text.	No
Supplemental Detail	Supplemental information about the event, including more detailed information than the message text.	No

The log file location depends on the type of component:

- For most Java components, the log file location is:

  ```
  (UNIX) DOMAIN_HOME/servers/server_name/logs
  (Windows) DOMAIN_HOME\servers\server_name\logs
  ```

  The default name of a log file is *server-name*-diagnostic.log.

- For system components, the default log file location is:

  ```
  (UNIX) DOMAIN_HOME/servers/component_name/logs
  (Windows) DOMAIN_HOME\servers\component_name\logs
  ```

Table 12-2 shows the log file location for components of Oracle Fusion Middleware.

Table 12-2 Log File Location for Oracle Fusion Middleware Components

Component	Log File Location
Fusion Middleware Control	*DOMAIN_HOME*/sysman/log/emoms.log *DOMAIN_HOME*/sysman/log/emoms.trc
Oracle Application Development Framework	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle BI Enterprise Edition	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log *DOMAIN_HOME*/servers/*instance_key*/logs/*server-name*-diagnostic.log
Oracle Enterprise Scheduler	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Event Processing	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Forms Services	*DOMAIN_HOME*/servers/*server_name*/logs/*application_name*.log
Oracle HTTP Server	*DOMAIN_HOME*/servers/*component_name*/logs/*.log
Oracle Managed File Transfer	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Platform Security Services	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle Service Bus	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle TopLink	*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
Oracle WebLogic Server	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle WebCenter Content	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log

Table 12-2 (Cont.) Log File Location for Oracle Fusion Middleware Components

Component	Log File Location
Oracle WebCenter Portal	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log
Oracle WebCenter Sites	*DOMAIN_HOME*/servers/*server_name*/logs/*server-name*-diagnostic.log *DOMAIN_HOME*/servers/*server_name*/logs/*sites_server-name*.log000*n* *DOMAIN_HOME*/servers/*server_name*/logs/*cas_server-name*.log000*n*
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*

Viewing and Searching Log Files

You can view, list, and search log files across Oracle Fusion Middleware components. You can view and search log files using Fusion Middleware Control or you can download a log file to your local client and view the log files using another tool. You can also list, view, and search log files using the WLST command-line tool.

Note the following about using the WLST commands to view the log files:

- 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, passing it the path to the domain home.
- Most of the WLST logging commands require that you are running in the domainRuntime tree. For example, to connect and to run in the domainRuntime tree, use the following WLST commands:

```
connect('username', 'password', 'localhost:port_number')
domainRuntime()
```

For more information about the commands, see Logging Custom WLST Commands in the *WLST Command Reference for Infrastructure Components*.

- Viewing Log Files and Their Messages
- Searching Log Files
- Downloading Log Files

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

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 domain. Right-click the Managed Server name and choose **Logs,** then **View Log Messages.**

   The Log Messages page is displayed.

2. Click **Target Log Files.**

   The Log Files page is displayed. On this page, you can see a list of log files related to the Managed Server.

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. By default, the messages are sorted by time, in ascending order. You can sort the messages by the any of the columns, such as Message Type, by clicking the column name.

Viewing Log Files and Their Messages Using WLST

You can list the log files for an Oracle WebLogic Server domain, a server, or component using the WLST `listLogs` command.

You can use this command while connected or disconnected. While connected, the default target is the Oracle WebLogic Server domain.

To list the log files, first use the `domainRuntime` command as described in Viewing and Searching Log Files . The following describes how to list and view log files:

- To list all of the log files for the Oracle WebLogic Server wls_server_1, use the following command:

```
listLogs(target='wls_server_1')
file://hostname/scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_1/logs/
wls_server_1.log
2017-03-21 06:55:37                500.1K wls_server_1.log00026
2017-03-21 07:49:08                500.1K wls_server_1.log00027
2017-03-21 08:46:29                500.4K wls_server_1.log00028
2017-03-21 09:45:29                500.4K wls_server_1.log00029
2017-03-21 10:43:00                500.3K wls_server_1.log00030
2017-03-21 11:39:56                500.3K wls_server_1.log00031
2017-03-21 12:38:56                500.4K wls_server_1.log00032
2017-03-21 13:18:06                358.1K wls_server_1.log

file://hostname/scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_1/logs/
wls_server_1.out
2017-03-13 11:00:05                   4M wls_server_1.out00001
2017-03-21 13:18:06                12.1M wls_server_1.out
...
```

- To list the logs for a system component, use one of the following formats:

```
listLogs(target='component_name')
listLogs(target='sc:component_name')
```

For example, to list the logs for the Oracle HTTP Server ohs1, use the following command:

```
listLogs(target='ohs1')
```

- To list the logs while disconnected, you must specify the `oracleInstance` parameter, passing it the path of the domain. For example, to list the log files for the Managed Server wls_server_1:

```
listLogs(oracleInstance='/scratch/Oracle/config/domains/WLS_domain',
         target='wls_server_1')
```

- To view the diagnostic messages in log files, use the WLST `displayLogs` command. This command works when you are either connected or disconnected.

  For example, to view the messages generated in the last 10 minutes in the log files for the Oracle WebLogic Server domain, use the following command:

```
displayLogs(last=10)

[2017-03-21T13:30:11.892-07:00] [wls_server_1] [WARNING] [WSM-09004]
 [oracle.wsm.resources.common] [host: hostname [nwaddr: 10.240.82.231]
[tid: [ACTIVE].ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-
tuning)'] [userId: OracleSystemUser]
[ecid: 66217af9-247f-4344-94a9-14f90e75a586-00070b85,0] [APP: wsm-pm] [TARGET: /
base_domain/wls_server_1/wsm-pm]
[LOG_FILE: /scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_1/logs/
wls_server_1-diagnostic.log]
Component auditing cannot be initialized.
[2017-03-21T13:30:11.895-07:00] [wls_server_1] [NOTIFICATION] [BEA-010227]
 [EJB] [host: hostname] [nwaddr: 10.240.82.231] [tid: [ACTIVE].ExecuteThread:
 '5' for queue: 'weblogic.kernel.Default (self-tuning)']
[userId: OracleSystemUser]
[ecid: 66217af9-247f-4344-94a9-14f90e75a586-00070b85,0] [TXN_ID:
BEA1-7438ECB7CDFCAF163A9A]
[TARGET: /base_domain/wls_server_1]
[LOG_FILE: /scratch/oracle1/Oracle/domains/base_domain/servers/wls_server_1/logs/
wls_server_1.log]
EJB exception occurred during invocation from home or business:
 weblogic.ejb.container.internal.StatelessEJBHomeImpl@314c2224 generated
 exception: java.lang.reflect.UndeclaredThrowableException
```

  The previous command returns the messages sorted by time, in ascending order.

- To display the logs for a system component, use one of the following formats:

```
listLogs(target='component_name')
listLogs(target='sc:component_name')
```

  For example, to display the log files for the Oracle HTTP Server ohs_1, use the following command:

```
displayLogs(target='sc:ohs_1')
```

You can search the messages by specifying particular criteria and sort the output, as described in Searching Log Files.

See Logging Custom WLST Commands in the *WLST Command Reference for Infrastructure Components* for more information about the `listLogs` and `displayLogs` commands.

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

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 topics describe how to search log files:

- Searching Log Files: Basic Searches
- Searching Log Files: Advanced Searches

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 10 minutes.
2. Click the **Search** icon. 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 10 minutes.
   - **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 12-3.
4. You can specify more search criteria.
5. Click **Search.**
6. To help identify messages of relevance, in the table, for **Show,** select **Messages**.

   You can also select how the messages are grouped, for example by host or incident ID.

   To view related messages, select a message, then click **View Related Messages** and select **by Time** or **by ECID (Execution Context ID).**

- **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.

Searching Log Files: Advanced Searches

You can refine your search criteria by clicking the **Search** icon and using the following additional controls:

- **Message:** Select an operator, such as **contains** and then enter a value to be matched.
- **Add Fields:** Click this to specify additional criteria, such as Host, which lets you narrow the search to particular hosts. Then click **Add.**

  For each field you add, select an operator, such as **contains** and then enter a value to be matched.

Searching Log Files Using WLST

You can search the log files using the WLST `displayLogs` command. You can narrow your search by specifying criteria, such as time, component ID, message type, or ECID. For example:

- To search for error messages generated in the last 5 minutes, for a system component such as the Oracle HTTP Server ohs1, use the following command:

  ```
  displayLogs(target='sc:ohs1', last=5)
  ```

- To search for error messages generated in the last 10 minutes for the Managed Server wls_server_1, use the following command:

  ```
  displayLogs(oracleInstance='/scratch/Oracle/config/domains/WLS_domain',
  target='wls_server_1', last=10)
  ```

You can narrow your search by using the `query` parameter and specifying criteria, such as component ID, message type, or ECID. In the `query` clause, you can specify a query expression with any of the attributes listed in Table 12-1. Some of the criteria you can use are:

- Types of messages. For example, to search for ERROR and INCIDENT_ERROR messages for the Managed Server wls_server_1, use the following command:

  ```
  displayLogs(oracleInstance='/scratch/Oracle/config/domains/wls_domain',
            target='wls_server_1',
            query='MSG_TYPE eq ERROR or MSG_TYPE eq INCIDENT_ERROR')
  ```

- A particular ECID. For example, to search for error messages with a particular ECID (0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0') for the Managed Server wls_server_1, use the following command:

  ```
  displayLogs(oracleInstance='/scratch/Oracle/config/domains/wls_domain',
           target='wls_server_1',
           query='ecid eq 0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0')
  ```

- Component type. For example, to search for messages from Oracle HTTP Server instances, use the following query:

  ```
  displayLogs(query='COMPONENT_ID eq ohs')
  ```

- Range of time. To search for error messages that occurred within a specified range of time, you specify the attribute TSTZ_ORIGINATING with both `from` and `to` operators, using the following format:

```
displayLogs(query='TSTZ_ORIGINATING from start_time and
                   TSTZ_ORIGINATING to end_time')
```

You specify the date using the following ISO 8601 time format:

```
YYYY-MM-DDThh:mm:ss-hh:mm_offset_from_UTC
```

For example:

```
2017-03-30T12:00:00:0000-08:00
```

For example, to display the error message from between 8:00 a.m. and 11 a.m. on March 17, 2017, use the following command:

```
displayLogs(query='TSTZ_ORIGINATING from 2017-03-17T08:00:00-07:00
            and TSTZ_ORIGINATING to 2017-03-17T11:00:00-07:00')
```

- Group messages. To display a count of messages, grouped by specific attributes, use the `groupBy` parameter to the WLST command `displayLogs`. For example, to display the count of WARNING messages by component, use the following command:

```
displayLogs(groupBy=['COMPONENT_ID'], query='MSG_TYPE eq WARNING')
```

- Group messages by supplemental attributes. If you use the DMS event tracing commands, you can create a destination that enables you to query and group messages by specific supplemental attributes. In this case, you use the addDMSEventDestination command to create a destination with the property writeDataAsMessageAttributes. (See addDMSEventDestination in the *WLST Command Reference for Infrastructure Components*.)

  Then, you can query the log messages. For example, to query by Completing Party:

```
displayLogs(log="DOMAIN_HOME/servers/AdminServer/logs/DMSEventTraceLoggerDestination-
event.log",
            groupBy=["SUPPL_ATTR.dms.NounType",
                     "SUPPL_ATTR.dms.NounPath",
                     "SUPPL_ATTR.org.service.CompletingParty"])
```

  This command returns the following:

```
-----------------------+-----------------+----------------------------+------
dms.NounType           | dms.NounPath    |org.service.CompletingParty|COUNT
-----------------------+-----------------+----------------------------+------
CallCenter_Agent       |/callAgent/Freya          | null              |    25
CallCenter_Agent       |/callAgent/Johann         | null              |    20
CallCenter_Agent       |/callAgent/Rhys           | null              |    25
CallCenter_City        |/callCenter/fr/Pau        | null              |     2
CallCenter_City        |/callCenter/fr/Vichy      | null              |     2
CallCenter_City        |/callCenter/uk/Watford    | null              |     2
CallCenter_Country     |/callCenter/de            | null              |     6
CallCenter_Country     |/callCenter/fr            | null              |     6
CallCenter_Country     |/callCenter/uk            | null              |     6
CallCenter_IncomingCall|/callCenter/fr/Pau/inCalls| agent             |    10
CallCenter_IncomingCall|/callCenter/fr/Pau/inCalls| caller            |    40
```

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 for Specific Components Using Fusion Middleware Control
- Downloading Specific Types of Messages Using Fusion Middleware Control

- Downloading Log Files Using WLST

Downloading Log Files Using Fusion Middleware Control

You can download the log messages to a file. You can download either the matching messages from a search or the messages in a particular log file.

To download the matching messages from a search to a file using Fusion Middleware Control:

1. From the navigation pane, expand the domain and select the target, for example by clicking on the domain.
2. From the dynamic target menu, such as the WebLogic Domain menu, choose **Logs,** then **View Log Messages.**

   The Log Messages page is displayed.
3. Search for particular types of messages as described in Searching Log Files Using Fusion Middleware Control.
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 File.** Click **OK.**

Downloading Log Files for Specific Components Using Fusion Middleware Control

To download the log files for a specific component using Fusion Middleware Control:

1. For system components, from the navigation pane, expand the installation type, such as **HTTP Server** and select the component. For Java components, from the navigation pane, expand the component type, and then select the component.
2. From the dynamic target menu, choose **Logs,** then **View Log Messages.**

   The Log Messages page is displayed.
3. Click **Target Log Files.**

   The Log Files page is displayed. On this page, you can see a list of log files related to the component or application.
4. Select a log file and click **Download.**
5. An Opening dialog box is displayed.
6. Select either **Open With** or **Save to Disk.** Click **OK.**

Downloading Specific Types of Messages Using Fusion Middleware Control

To export specific types of messages or messages with a particular Message ID to a file:

1. From the navigation pane, expand the domain and select a target.
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 Searching Log Files Using Fusion Middleware Control.
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 **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.
7. Select either **Open With** or **Save to Disk.** Click **OK.**

Downloading Log Files Using WLST

You can download log files using the WLST `displayLogs` command and redirecting the output to a file. For example:

```
displayLogs(type=['ERROR','INCIDENT_ERROR'], exportFile='/scratch/tmp/download_log.txt')
```

The messages are written to the file download_log.txt in the specified directory. By default, they are written to standard output.

Configuring Settings for Log Files

You can change the location of log files, how often the log files rotate, and the level of information written to them, along with other configuration settings. You can change the log settings of Managed Servers and Java components using Fusion Middleware Control or WLST.

> **Note:**
>
> For many system components, which are listed in Using WLST Commands with System Components, you cannot configure settings for log files using Fusion Middleware Control. For information about how to configure options for log files for system components, see the administrator's guide for the component.

Note the following about using the WLST commands to configure log settings:

- The configuration commands, such as `setLogLevel`, only work in connected mode. That is, you must connect to a running WebLogic Server instance before you invoke the commands.

  The configuration commands are supported for Java components that run within a WebLogic Server, but are not supported for Oracle WebLogic Server. The configuration commands are not supported for system components.
- Most of the WLST logging commands require that you are running in the domainRuntime tree. For example, to connect and to run in the domainRuntime tree, use the following commands:

```
connect('username', 'password', 'localhost:port_number')
domainRuntime()
```

- The `listLoggers`, `getLogLevel`, and `setLogLevel` commands work in `config` and `runtime` mode. In `config` mode the commands work on loggers that are defined in the configuration file. In `runtime` mode, the commands work directly with loggers that are defined in the server JVM. By default, the `setLogLevel` command sets the level on the run-time logger and updates the logger definition in the configuration file. By default, the `listLoggers` and `getLogLevel` commands return run-time loggers.

For more information about these commands, see Logging Custom WLST Commands in the *WLST Command Reference for Infrastructure Components*.

For Java components, you can configure the names and locations of log files, the size of the log files, the level of information written to the log files, the format, and the Locale encoding, as described in the following topics:

- Changing Log File Locations
- Configuring Log File Rotation
- Setting the Level of Information Written to Log Files
- Specifying the Log File Format
- Specifying the Log File Locale

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

Changing Log File Locations Using Fusion Middleware Control

To change the name and location of a component log file using Fusion Middleware Control:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs,** then **Log Configuration.**

   The Log Configuration page is displayed.

   Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration,** then **Log Configuration.**
3. Select the Log Files tab.
4. In the table, select the log handler and click **Edit.**
5. For **Log Path,** enter a new path.
6. Click **OK.**
7. In the confirmation window, click **Close.**

Note that if you change the location of Oracle HTTP Server log files, the location of the access_log and ohs*n*.log files are changed, but the location of console~OHS~1.log is not changed.

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')
```

Configuring Log File Rotation

An **ODL log** is a set of log files that includes the current ODL log file and zero or more **ODL Archives (segment files)** that contain older messages. As the log file grows, new information is added to the end of the log file, *server_name*`-diagnostic.log`. When the log file reaches the rotation point, it is renamed and a new log file, *server_name*`-diagnostic.log` is created. You specify the rotation point, by specifying the maximum ODL segment size or the rotation time and rotation frequency.

Segment files are created when the ODL log file *server_name*`-diagnostic.log` reaches the rotation point. That is, the *server_name*`-diagnostic.log` is renamed to *server_name*`-diagnostic-`*n*`.log`, where *n* is an integer, and a new *server_name*`-diagnostic.log` file is created when the component generates new diagnostic messages.

To limit the size of the ODL log, you can specify:

- The maximum size of the logging directory. Whenever the sum of the sizes of all of the files in the directory reaches the maximum, the oldest archive is deleted to keep the total size under the specified limit.

  By default, the log files are rotated when they reach 10 MB. The maximum size of all log files for a particular component is 100 MB.
- The maximum size of the log file. You specify that a new log file be created when a specific time or frequency is reached.

> **Note:**
>
> After you change the log file rotation, the configuration is reloaded dynamically. It may take 1 or 2 seconds to reload the configuration.

The following topics describe how to change the rotation:

- Specifying Log File Rotation Using Fusion Middleware Control
- Specifying Log File Rotation Using WLST

Specifying Log File Rotation Using Fusion Middleware Control

To configure log file rotation using Fusion Middleware Control:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs,** then **Log Configuration.**

   The Log Configuration page is displayed.

   Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration,** then **Log Configuration.**

3. Select the Log Files tab.
4. In the table, select the logger and click **Edit.**

   The Edit Log File dialog box is displayed.
5. In the Rotation Policy section, you can select one of the following:
   - **Size Based:** If you select this, enter the following:
     - For **Maximum Log File Size,** enter the size in MB, for example, 15.
     - For **Maximum Size of All Log Files,** enter the size in MB, for example, 150.
   - **Time Based:** If you select this, enter the following:
     - For **Start Time,** click the calendar and select the date and time when you want the rotation to start. For example, select September 8, 2010 6:00 AM.
     - For **Frequency,** you can select **Minutes** and enter the number of minutes, or you can select **Hourly, Daily,** or **Weekly.**
     - For **Retention Period,** you can specify how long the log files are kept. You can select **Minutes** and enter the number of minutes, or you can specify **Day, Week**, **Month,** or **Year.**

       Specifying a shorter period means that you use less disk space, but are not able to retrieve older information.
6. Click **OK.**
7. In the confirmation window, click **Close.**

Specifying Log File Rotation Using WLST

To specify log file rotation using WLST, use the `configureLogHandler` command. You can specify size-based rotation or time-based rotation.

For example, to specify that the log files rotate daily and that they are retained for a week, use the following command:

```
configureLogHandler(name='odl-handler', rotationFrequency='daily',
                    retentionPeriod='week')
```

To specify that the size of a log file does not exceed 5 MB and rotates when it reaches that size, use the following command:

```
configureLogHandler(name='odl-handler', maxFileSize='5M')
```

Setting the Level of Information Written to Log Files

You can configure the amount and type of information written to log files by specifying the message type and level. For each message type, possible values for the message level are from 1 (lowest severity) through 32 (highest severity). Some components support only some of the levels for each message type. See the administrator's guide for your component for more information. Generally, you need to specify only the type; you do not need to specify the level.

When you specify the type, Oracle Fusion Middleware returns all messages of that type, as well as the messages that have a higher severity. For example, if you set the message type to WARNING, Oracle Fusion Middleware also returns messages of type INCIDENT_ERROR and ERROR.

Table 12-3 describes the message types and the most common levels for each type.

Table 12-3 Diagnostic Message Types and Level

Message Type	Level	Description
INCIDENT_ERROR	1	A serious problem that may be caused by a bug in the product and that should be reported to Oracle Support. Examples are errors from which you cannot recover or serious problems.
ERROR	1	A serious problem that requires immediate attention from the administrator and is not caused by a bug in the product. An example is if Oracle Fusion Middleware cannot process a log file, but you can correct the problem by fixing the permissions on the document.
WARNING	1	A potential problem that should be reviewed by the administrator. Examples are invalid parameter values or a specified file does not exist.
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.
NOTIFICATION	32	The finest 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 and 32: 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 12-4 shows the log level mappings among ODL format, Oracle WebLogic Server, and Java.

Table 12-4 Mapping of Log Levels Among ODL, Oracle WebLogic Server, and Java

ODL	WebLogic Server	Java
OFF	OFF	2147483647 - OFF
INCIDENT_ERROR:1	(EMERGENCY)	1100

Table 12-4 (Cont.) Mapping of Log Levels Among ODL, Oracle WebLogic Server, and Java

ODL	WebLogic Server	Java
INCIDENT_ERROR:4	EMERGENCY	1090
INCIDENT_ERROR:14	ALERT	1060
INCIDENT_ERROR:24	CRITICAL	1030
ERROR:1	(ERROR)	1000 - SEVERE
ERROR:7	ERROR	980
WARNING:1	WARNING	900 - WARNING
WARNING:7	NOTICE	880
NOTIFICATION:1	INFO	800 - INFO
NOTIFICATION:16	(DEBUG)	700 - CONFIG
TRACE:1	(DEBUG)	500 - FINE
TRACE:1	DEBUG	495
TRACE:16	(TRACE)	400 - FINER
TRACE:32	(TRACE)	300 - FINEST
TRACE:32	TRACE	295

You can configure the message levels written to a log file for a particular log file or a logger using Fusion Middleware Control or WLST commands, as described in the following topics:

- Configuring Message Levels for a Log File Using Fusion Middleware Control
- Configuring Message Levels for Loggers Using Fusion Middleware Control
- Configuring Message Levels Using WLST

Configuring Message Levels for a Log File Using Fusion Middleware Control

To set the message level for a component log file:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs,** then **Log Configuration.**

   The Log Configuration page is displayed.

   Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration,** then **Log Configuration.**
3. Select the Log Files tab.
4. In the table, select the log file and click **Edit.**

   The Edit Log File dialog box is displayed.
5. For **Log Level,** select the logging level. For example, select **WARNING:1 (WARNING).**
6. Click **OK.**
7. In the confirmation window, click **Close.**

Configuring Message Levels for Loggers Using Fusion Middleware Control

To set the message level for one or more loggers for a component:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs,** then **Log Configuration.**

   The Log Configuration page is displayed.

   Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration,** then **Log Configuration.**
3. Select the **Log Levels** tab.
4. For **View,** select **Runtime Loggers** or **Loggers with Persistent Log Level State.**

   Run-time loggers are loggers that are currently active. Persistent loggers are loggers that are saved in a configuration file and the log levels of these loggers are persistent across component restarts. A run-time logger can also be a persistent logger, but not all run-time loggers are persistent loggers.
5. In the table, to specify the same level for all loggers, select the logging level for the top-level logger. Then, for child loggers that do not specify that the logging level is inherited from the parent, specify **Inherited from Parent.** For most situations, that is sufficient.

   However, if you need to specify the level for a particular logger, expand the logger and then, for the logger that you want to modify, select the logging level. For example, for the logger oracle.wsm.management.logging, select **WARNING:1 (WARNING).**
6. Click **Apply.**

Configuring Message Levels Using WLST

To set the message level with WLST, you use the `setLoglevel` command. To get the current message level, you use the `getLogLevel` command. You must be connected to WebLogic Server before you use the configuration commands.

You can view the log level for a logger for an Oracle WebLogic Server. For example, to view the log level of the Oracle WebLogic Server wls_server_1, use the following command:

```
getLogLevel(logger='oracle',  target='wls_server_1')

NOTIFICATION:1
```

You can set the log level for a particular logger. The following example sets the message type to WARNING for the logger oracle.wsm.msg.logging:

```
setLogLevel(target='wls_server_1', logger='oracle.wsm.msg.logging', level='WARNING')
```

To get a list of loggers for the Oracle WebLogic Server wls_server_1, use the listLoggers command:

```
listLoggers(target='wls_server_1')
.
.
.
oracle.wsm.msg.logging                                  | NOTIFICATION:1
oracle.wsm.nobehavior.model.NoBehaviorAssertion         | <Inherited>
oracle.wsm.policy.advertisement.AdvertisementContext    | <Inherited>
oracle.wsm.policy.model.impl.AndCompositeAssertion      | <Inherited>
.
```

.
.

You can also filter logger names using the pattern parameter and a regular expression. For example, to return all loggers that begin with `oracle` in the Oracle WebLogic Server wls_server_1, use the following command:

```
listLoggers(target='wls_server_1', pattern='oracle.*')

--------------------------------------------------------------------------------
Logger                                                 | Level
--------------------------------------------------------------------------------
oracle                                                   NOTIFICATION:1
 oracle.adf                                              <Inherited>
oracle.adf.controller                                    <Inherited>
oracle.adf.desktopintegration                            <Inherited>
oracle.adf.faces                                         <Inherited>
```

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

Specifying the Log File Format Using Fusion Middleware Control

To change the format using Fusion Middleware Control:

1. From the navigation pane, select the component.
2. From the dynamic target menu, choose **Logs,** then **Log Configuration.**

   The Log Configuration page is displayed.

   Note that the navigation may be different for some components. For example, for Oracle HTTP Server, you choose **Administration,** then **Log Configuration.**
3. Select the Log Files tab.
4. In the table, select the log file and click **Edit.**

   The Edit Log File dialog box is displayed.
5. For Log File Format, select **Oracle Diagnostics Logging - Text** or **Oracle Diagnostics Logging - XML.**
6. Click **OK.**
7. In the confirmation window, click **Close.**

Specifying the Log File Format Using WLST

To specify the log file format using WLST, you use the `configureLogHandler` command, with the `format` parameter and specify either ODL-Text or ODL-XML. ODL-Text is the default.

For example, to specify ODL-XML format, use the following command:

```
configureLogHandler(name='odl-handler', format='ODL-XML')
```

Specifying the Log File Locale

The language and data formats used in the log files are determined by the default locale of the server Java Virtual Machine (JVM). You can change them using the Language and Regional Options applet in Control Panel on Windows or the LANG and LC_ALL environment variables on a UNIX platform.

The character encoding of log files is determined by the server JVM's default character encoding or an optional configuration setting. You should choose an encoding that supports all languages used by the users, or the log file may be corrupted. By default, the log is in the server JVM's default character encoding. If you change the encoding, delete or rename old log files to prevent them from being damaged by the new logs appended in a different encoding.

For support of any language, Oracle recommends that you use Unicode UTF-8 encoding. On a UNIX operating system, setting the LANG and LC_All environment variables to a locale with the UTF-8 character set enables UTF-8 logging (for example, `en_US.UTF-8` for the US locale in UTF-8 encoding).

You can specify the log file locale using WLST commands or by editing a file, as described in the following topics:

- Specifying the Log File Encoding Using WLST
- Specifying the Log File Encoding in logging.xml

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")
```

Specifying the Log File Encoding in logging.xml

To specify the log file encoding in the logging.xml file, use an optional encoding property to specify the character set encoding.

The logging.xml file is located in the following directory:

DOMAIN_HOME`/config/fmwconfig/servers/`*server_name*`/`

For example, to specify UTF-8, add the following encoding property in the log_handler element:

```
<property name='encoding' value='UTF-8'/>
```

About 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.

This section contains the following topics:

- Understanding ECIDs and RIDs in Correlating Messages

- Correlating Messages Across Messages and Components

Understanding ECIDs and RIDs in Correlating Messages

Each diagnostic message contains an **Execution Context ID (ECID)** and a **Relationship ID (RID)**:

- An ECID is a globally unique identifier associated with the execution of a particular request. An ECID is generated when the request is first processed.
- A RID distinguishes the work done in one thread on one process, from work done by any other threads on this and other processes on behalf of the same request.

The ECID and RID help you to use log file entries to correlate messages from one application or across Oracle Fusion Middleware components. By searching for related messages using the message correlation information, multiple messages can be examined and the component that first generates a problem can be identified (this technique is called **first-fault component isolation**). Message correlation data can help establish a clear path for a diagnostic message across components, within which errors and related behavior can be understood.

You can use the ECID and RID to track requests as they move through Oracle Fusion Middleware.

The following shows an example of an ECID:

```
0000I3K7DCnAhKB5JZ4Eyf19wAgN000001,0
```

The RID is one or more numbers separated by a colon (:). The first RID created for a request is 0. Each time work is passed from a thread that has an ECID associated with it to another thread or process, a new RID is generated that encodes the relationship to its creator. That is, a new generation is created. Each shift in generation is represented by a colon and another number. For example, the seventh child of the third child of the creator of the request is:

```
0:3:7
```

Correlating Messages Across Messages and Components

You can view all the messages with the same ECID using the WLST `displayLogs` command. The following example searches for the ECID in the domain:

```
displayLogs(ecid='0000Hl9TwKUCslT6uBi8UH181kWX000002')
```

You can also search for the ECID in a WebLogic Server instance, or a system component, by specifying it in the target option.

You can search for messages with a particular ECID on the Log Messages page in Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Logs,** then **View Log Messages.**

   To search for messages for a component or application, select the component or application and then choose **Logs,** then **View Log Messages** from that target's menu.
2. Specify search criteria, as described in Searching Log Files: Advanced Searches.
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.

5. Trace the ECID to the earliest message. (You may need to increase the scope to view the first message with the ECID.)

Configuring Tracing

Sometimes you need more information to troubleshoot a problem than is usually recorded in the logs. One way to achieve that is to increase the level of messages logged by one or more components, and fine-tune which messages are written to the log files.

For example, you can set the logging level to TRACE:1 or TRACE:32, as described in Setting the Level of Information Written to Log Files, which results in more detailed messages being written to the log files. This is referred to as **tracing**.

However, tracing can often result in a large amount of log messages being written to the log files. Oracle Fusion Middleware provides the following mechanisms to fine-tune which messages are traced:

- QuickTrace, which provides fine-grained logging to memory
- Selective Trace, which provides fine-grained logging for a specific user or other properties of a request

The following topics provide information about how to use these mechanisms:

- Configuring and Using QuickTrace
- Configuring and Using Selective Tracing

Configuring and Using QuickTrace

QuickTrace provides fine-grained logging to memory. The following topics describe Quick Trace and how to enable and use it:

- About Quick Trace
- Configuring QuickTrace
- Writing Trace Messages to a File
- Disabling QuickTrace Using WLST

About Quick Trace

With QuickTrace, you can trace messages from specific loggers and store the messages in memory. Because QuickTrace logs the messages to memory, it avoids the cost of formatting, string manipulations, and input/output operations. As as result, you can enable fine-level application logging for specific loggers without performance overhead.

By default, QuickTrace writes the messages to one common buffer. However, you can specify that messages for particular users are written to separate buffers.

You can save the messages that are in memory to a file by invoking the QuickTrace Dump in Fusion Middleware Control as described in Writing the Trace Messages to a File Using Fusion Middleware Control or by using the WLST, as described in Writing the Trace Messages to a File Using WLST.

To enable QuickTrace, you create a QuickTrace handler and associate a logger with it. You can specify the buffer size, as well as other attributes, for the handler. Then, you set the level of the amount and type of information to be written by the loggers to memory.

Configuring QuickTrace

You can configure and use QuickTrace using Fusion Middleware Control or WLST, as described in the following topics:

- Configuring QuickTrace Using Fusion Middleware Control
- Configuring QuickTrace Using WLST

Configuring QuickTrace Using Fusion Middleware Control

To configure QuickTrace using Fusion Middleware Control:

1. From the navigation pane, expand the domain. Right-click the Managed Server name and choose **Logs,** then **Log Configuration.**

   The Log Configuration page is displayed.
2. Select the QuickTrace tab.
3. Click **Create.**

   The Create QuickTrace Handler dialog box is displayed.
4. For **Name,** enter a name for the handler.
5. For **Buffer Size,** enter the size, in bytes, for the buffer for storing log messages in memory. The default is 5242880.
6. For **Maximum Field Length,** enter the length, in bytes, for each field in a message. The fields can include the message text, supplemental attributes, and the thread name. The default is 240.

   An excessively long field for each message can reduce the amount of log records in the buffer.
7. For **Handler Level,** select the log level for the handler. See Setting the Level of Information Written to Log Files for information about the levels.
8. For **Loggers to Associate,** select the loggers that you want to associate with this QuickTrace handler. All messages of the specified level for these handlers will be written to memory.

   Many loggers are associated with other handlers. For example, the oracle.adf logger is associated with the handlers odl-handler, wls-domain, and console-handler. When you set the level of the logger, these handlers will use the same level, such as TRACE:1, for the logger, such as oracle.adf. As a result, much information will be written to the log files, consuming resources. To avoid consuming resources, set the level of the handlers to a lower level, such as WARNING or INFORMATION.
9. Select **Enable User Buffer?** if you want to enable a user buffer. If you enable this, the handler maintains an individual buffer for each user you specify.

   Then, for **User Names for Reserve Buffer,** enter the names of the users, separated by commas.
10. For the remaining options, accept the default values. For information about the options, see ConfigureLogHandler in the *WLST Command Reference for Infrastructure Components*.
11. Click **OK.**
12. When the configuration completes processing, click **OK.**

Now, messages of the specified level for the specified loggers are written to memory.

Configuring QuickTrace Using WLST

To configure QuickTrace using WLST, you associate a logger with the QuickTrace handler, using the configureLogHandler command.

For example, to associate the oracle.adf logger with the QuickTrace handler and write all TRACE:1 messages to memory:

1. Use the configureLogHandler command to associate the logger with the QuickTrace handler:

```
configureLogHandler(name="quicktrace-handler", addToLogger="oracle.adf")

Handler Name: quicktrace-handler
type:   oracle.core.ojdl.logging.QuickTraceHandlerFactory
encoding:       UTF-8
maxFieldLength: 240
mode:   objRef
useThreadName:  false
useSourceClassandMethod:        false
useLoggingContext:      false
bufferSize:     5242880
```

   The messages for the handler are written to a common buffer.

   You can set additional properties for the QuickTrace handler. For example, to enable user buffers for the users user1 and user2:

```
configureLogHandler(name="quicktrace-handler", addToLogger="oracle.adf.faces",
      propertyName="enableUserBuffer", propertyValue="true",
      propertyName="enableUserBuffer", propertyValue="user1, user2")
...
Handler Name: quicktrace-handler
type:   oracle.core.ojdl.logging.QuickTraceHandlerFactory
useLoggingContext:      false
bufferSize:     5242880
 .
 .
 .
reserveBufferUserID:    user1, user2
enableUserBuffer:       true
```

   Messages for user1 and user2 are written to separate buffers. In addition, messages related to other users are written to the common buffer.

   To confirm the settings for the handler, use the listLogHandlers command, as described in listLogHandlers in the *WLST Command Reference for Infrastructure Components*.

2. Set the level of the logger, using the setLogLevel command:

```
setLogLevel(logger='oracle.adf', level='TRACE:1')
```

   To confirm the settings for the logger, use the listLoggers command, as described in listLoggers in the *WLST Command Reference for Infrastructure Components*.

3. Many loggers are associated with other handlers. For example, the oracle.adf logger is associated with the handlers odl-handler, wls-domain, and console-handler. When you set the level of the logger, these handlers will use the same level (TRACE:1) for the logger oracle.adf. As a result, much information will be written to the log files, consuming resources. To avoid consuming resources, set the level of the handlers to a lower level, such as WARNING or INFORMATION.

For this example, set the level of the three handlers to WARNING:1:

```
configureLogHandler(name="odl-handler", level="WARNING:1")
configureLogHandler(name="wls-domain", level="WARNING:1")
configureLogHandler(name="console-handler", level="WARNING:1")
```

Note that you should keep the level of the QuickTrace handler at ALL, which is the default.

See configureLogHandler in the *WLST Command Reference for Infrastructure Components*

To confirm the level for the handler, use the getLogLevel command, as described in Configuring Message Levels Using WLST.

Writing Trace Messages to a File

You can write trace messages to a file using Fusion Middleware Control or WLST, as described in the following topics:

- Writing the Trace Messages to a File Using Fusion Middleware Control
- Writing the Trace Messages to a File Using WLST

Writing the Trace Messages to a File Using Fusion Middleware Control

You can save the messages that are in memory to a file by invoking the QuickTrace Dump in Fusion Middleware Control:

1. From the QuickTrace tab of the Log Configuration page, select the handler and click **Invoke QuickTrace Dump.**

   The Invoke QuickTrace Dump dialog box is displayed.

2. For **Buffer Name,** if you have specified user buffers when you configured the QuickTrace handler, select the user, or select Common Buffer for users that you did not specify. If you did not specify any user buffers, Common Buffer is the only option.

3. Click **OK.**

   When the processing is complete, the View Log Messages page is displayed.

4. You can search the messages, as described in Searching Log Files, and you can correlate the messages as described in About Correlating Messages Across Log Files and Components.

   In addition, you can download the messages to a file, as described in Downloading Log Files Using Fusion Middleware Control.

Writing the Trace Messages to a File Using WLST

You can save the messages to a file by using the executeDump command.

For example:

```
executeDump(name="odl.quicktrace", outputFile="/scratch/oracle1/qt1.dmp")
```

The command writes the dump to the specified file.

For more information about the executeDump command, see Executing Dumps.

In addition, if an incident is created (automatically or manually), the QuickTrace messages are written to dump files in the incident directory. If you enabled user buffers, each user will have one file and the common buffer will have one file.

The file names have the following format:

```
odl_quicktraceN_iincident_number.username.dmp
```

For example:

```
odl_quicktrace6_i1.weblogic.dmp
```

See Creating an Incident Manually for information about creating an incident.

Disabling QuickTrace Using WLST

To disable QuickTrace, use the WLST configureLogHandler command and specify that the level is OFF:

```
configureLogHandler(name="quicktrace-handler", level="OFF")

Handler Name: quicktrace-handler
type:   oracle.core.ojdl.logging.QuickraceHandlerFactory
 .
 .
 .
reserveBufferUserID:    user1, user2
enableUserBuffer:       true
```

To remove a specific logger from association with the QuickTrace handler, use the configureLogHandler command with the removeFromLogger parameter:

```
configureLogHandler(name="quicktrace-handler", removeFromLogger="oracle.adf.faces")

Handler Name: quicktrace-handler
type:   oracle.core.ojdl.logging.QuickraceHandlerFactory
reserveBufferUserID:    user1, user2
enableUserBuffer:       true
```

See configureLogHandler in the *WLST Command Reference for Infrastructure Components* for complete syntax.

Configuring and Using Selective Tracing

Selective tracing provides fine-grained logging for specified users or other attributes of a request.

The following topics describe selective tracing and how to manage it using Fusion Middleware Control or WLST:

- About Selective Tracing
- Configuring Selective Tracing
- Viewing Selective Traces
- Disabling Selective Tracing

About Selective Tracing

Selective tracing provides fine-grained logging for specified users or other attributes of a request.

For example, a user cannot perform some functions because of security permissions, but it is not clear what operations or lack of permission for those operations are posing a problem.

In this case, you can enable tracing across the entire system but this would generate a large volume of log messages for all users in the system, not only for the user having a problem. With selective tracing, you can enable tracing only for the user who is having a problem. Then, you can ask the user to retry the functions. Following that, you can look at the trace messages which apply to the specific request made by the user.

You can also specify the logger to narrow the scope of the messages being logged.

Configuring Selective Tracing

You can configure selective tracing using Fusion Middleware Control or WLST, as described in the following topics:

- Configuring Selective Tracing Using Fusion Middleware Control
- Configuring Selective Tracing Using WLST

Configuring Selective Tracing Using Fusion Middleware Control

To configure selective tracing using Fusion Middleware Control:

1. From the navigation pane, right-click the domain name and choose **Logs,** then **Selective Tracing.**

   The Selective Tracing page is displayed.
2. For **Application Name,** select an application.
3. To add more fields, click **Add Fields** and select one of the options, such as Client Host or User Name.
4. For **Level,** select a logging level. Table 12-3 describes the logging levels.
5. For **Description,** enter a description.
6. For **Duration,** enter the number of minutes that you want the selective trace to run.

   The selective trace is disabled after the specified time.
7. For Trace ID, select either **Generate a New Unique Trace ID** or **Use a Custom Trace ID.** If you select Use a Custom Trace ID, enter an ID of your choosing, but make sure that it is unique. Note Fusion Middleware Control does not verify the uniqueness of the ID.
8. In the ODL section, select **Enable.**
9. In the DMS section, select **Enable.**
10. In the Loggers section, by default, all loggers are selected.

    You can select specific loggers that you want to trace. To find particular loggers, you can enter a string in the field above the table and click the Return key. For example, to find all loggers that begin with oracle.security, enter oracle.security.

    Then, in the table, select the loggers in the **Enable on All Servers** column.

    Note when you select loggers, those loggers apply to all current and active traces. Also note that even if you disable the loggers, you may see messages because all loggers have a general logging level, such as Notification. Those messages would still be written.
11. Click **Start Tracing.**

Now that you have started the trace, you can view active traces, as well as former traces, as described in Viewing Selective Traces Using Fusion Middleware Control.

Configuring Selective Tracing Using WLST

You can configure loggers for selective tracing and start tracing using the WLST configureTracingLoggers and startTracing commands.

For the simplest case, you can configure and start a trace using the startTracing command. When you do so, the selective tracing includes all loggers enabled for selective tracing.

For example, user1 receives errors when attempting to perform certain operations. To start a trace of messages related to user1 and to set the logging level to FINE, use the following command:

```
startTracing(user="user1",level="FINE")
Started tracing with ID: 885649f7-8efd-4a7a-9898-accbfc0bbba3
```

The startTracing command does not provide options to include or exclude particular loggers. In this case, you can use the configureTracingLoggers command. This command allows you to configure selective tracing to include only particular loggers and particular Oracle WebLogic Server instances. Note that the options you specify apply to all current and active traces.

For example, to configure selective tracing to include only security-related loggers:

1. Specify that all loggers be disabled for tracing, as shown in the following example:

   ```
   configureTracingLoggers(action="disable")
   Configured 1244 loggers
   ```

2. Enable the security-related loggers, by specifying the pattern option with a regular expression:

   ```
   configureTracingLoggers(pattern='oracle.security.*', action="enable")
   Configured 62 loggers
   ```

   To see a list of the loggers that support selective tracing, use the WLST listTracingLoggers command, as shown in the following example:

   ```
   listTracingLoggers(pattern="oracle.security.*")
   ------------------------------------------------------------------+--------
   Logger                                                            | Status
   ------------------------------------------------------------------+--------
   oracle.security                                                   | enabled
   oracle.security.audit.logger                                      | enabled
   oracle.security.jps.az.common.util.JpsLock                        | enabled
    .
    .
    .
   ```

3. Use the startTracing command, specifying the users and the level. For example:

   ```
   startTracing(user="user1",level="FINE")
   Started tracing with ID: a9580e65-13c4-420b-977e-5ba7dd88ca7f
   ```

See the following commands in the *WLST Command Reference for Infrastructure Components* for complete syntax:

- configureTracingLoggers
- startTracing
- listTracingLoggers

Viewing Selective Traces

You can view selective traces using Fusion Middleware Control or WLST, as described in the following topics:

- Viewing Selective Traces Using Fusion Middleware Control
- Viewing Selective Traces Using WLST

Viewing Selective Traces Using Fusion Middleware Control

You can view the selective traces that are currently active and the history of selective traces.

To view the selective traces:

1. From the Selective Tracing page, select the **Active Traces and Tracing History** tab.

   The tab shows a table with the active traces and a table with the tracing history.

2. To view a trace, select it from the appropriate table.

   The Log Messages page is displayed, with the messages that were captured by Selective Tracing. You can search the messages, as described in Searching Log Files, and you can correlate the messages as described in About Correlating Messages Across Log Files and Components.

   In addition, you can download the messages to a file, as described in Downloading Log Files Using Fusion Middleware Control.

Viewing Selective Traces Using WLST

After you have begun a trace, you can see the active traces by using the listActiveTraces command, as shown in the following example:

```
listActiveTraces()

------------------------------------+----------+-----------+------+--------------+-----------
Trace ID                            |Attr. Name|Attr. Value| Level| Exp. Time    |Description
------------------------------------+----------+-----------+------+--------------+-----------
b73b351c-9a9b-47df-b05a-356a336d5780 | USER_ID  | user1     | FINE | 5/22/17 11:17 AM |
a9580e65-13c4-420b-977e-5ba7dd88ca7f |USER_ID   |user1      | FINE | 5/22/17 11:19 AM |
```

You can view the contents of the trace using the displayLogs command and passing it the trace ID. You can also view traces that have stopped. For example:

```
displayLogs("a9580e65-13c4-420b-977e-5ba7dd88ca7f")
```

See listActiveTraces in the *WLST Command Reference for Infrastructure Components* for complete syntax.

Disabling Selective Tracing

You can configure selective tracing, view traces, and disable selective tracing using WLST, as described in the following topics:

- Disabling Selective Tracing Using Fusion Middleware Control
- Disabling Selective Traces Using WLST

Disabling Selective Tracing Using Fusion Middleware Control

To disable selective tracing using Fusion Middleware Control:

1. From the navigation pane, right-click the domain name and choose **Logs,** then **Selective Tracing.**
2. Select the **Active Traces and Tracing History** tab.
3. In the Active Traces table, select the trace and click **Disable.**

Disabling Selective Traces Using WLST

To avoid excessive logging in the system, you can disable a selective trace when you have obtained the information that you need. To disable a selective trace, you use the WLST stopTracing command, passing it the trace ID or user. For example:

```
stopTracing(traceId="885649f7-8efd-4a7a-9898-accbfc0bbba3")
Stopped 1 traces
```

You can also disable all traces by using the stopAll option. For example:

```
stopTracing(stopAll=1)
```

See stopTracing in the *WLST Command Reference for Infrastructure Components* for complete syntax.

13

Diagnosing Problems

The Oracle Fusion Middleware Diagnostic Framework helps you to collect and manage information about a problem so that you can resolve it or send it to Oracle Support for resolution.

- About 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.
- 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.
- Configuring the Diagnostic Framework
  You can configure some settings for the Diagnostic Framework, including custom diagnostic rules and problem suppression. In addition, you can configure an WLDF Policies and Actions to create an incident.
- Investigating, Reporting, and Solving a Problem
  You can use WLST and ADRCI commands and the Remote Diagnostic Agent (RDA) to investigate and report a problem (critical error), and in some cases, resolve the problem.

About 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, the Diagnostic Framework assigns it an incident number, and diagnostic data for the error (such as log files) are immediately captured and tagged with this number. The data is then stored in the Automatic Diagnostic Repository (ADR), where it can later be retrieved by incident number and analyzed.

The goals of the Diagnostic Framework are:

- First-failure diagnosis
- Limiting damage and interruptions after a problem is detected
- Reducing problem diagnostic time
- Reducing problem resolution time
- Simplifying customer interaction with Oracle Support

The Diagnostic Framework includes the following technologies:

- **Automatic capture of diagnostic data upon first failure:** For critical errors, the ability to capture error information at first failure greatly increases the chance of a quick problem resolution and reduced downtime. The Diagnostic Framework automatically collects

diagnostics, such as thread dumps, DMS metric dumps, and WebLogic Diagnostics Framework (WLDF) server image dumps. Such diagnostic data is similar to the data collected by airplane "black box" flight recorders. When a problem is detected, alerts are generated and the fault diagnosability infrastructure is activated to capture and store diagnostic data. The data is stored in a file-based repository and is accessible with command-line utilities.

- **Standardized log formats:** Standardized log formats (using the ODL log file format) across all Oracle Fusion Middleware components allows administrators and Oracle Support personnel to use a single set of tools for problem analysis. Problems are more easily diagnosed, and downtime is reduced.
- **Diagnostic rules:** Each component defines diagnostic rules that are used to evaluate whether a given log message should result in an incident being created and which dumps should be executed. The diagnostic rules also indicate whether an individual dump should be created synchronously or asynchronously.

  In addition, you can define custom rules that apply to a domain, a server, or an application in a domain or server.
- **Incident detection log filter:** The incident detection log filter implements the java.util.logging filter. It inspects each log message to see if an incident should be created, basing its decision on the diagnostic rules for components and applications.
- **Incident packaging service (IPS) and incident packages:** The IPS enables you to automatically and easily gather the diagnostic data—log files, dumps, reports, and more—pertaining to a critical error that has a corresponding incident, and package the data into a zip file for transmission to Oracle Support. All diagnostic data relating to a critical error that has been detected by the Diagnostics Framework is captured and stored as an incident in ADR. The incident packaging service identifies the required files automatically and adds them to the zip file.

  Before creating the zip file, the IPS first collects diagnostic data into an intermediate logical structure called an incident **package.** Packages are stored in the Automatic Diagnostic Repository. If you choose to, you can access this intermediate logical structure, view and modify its contents, add or remove additional diagnostic data at any time, and when you are ready, create the zip file from the package and upload it to Oracle Support.
- **Integration with WebLogic Diagnostics Framework (WLDF):** The Oracle Fusion Middleware Diagnostics Framework integrates with some features of WebLogic Diagnostics Framework (WLDF), including the capturing of WebLogic Server images on detection of critical errors. WLDF is a monitoring and diagnostic framework that defines and implements a set of services that run within WebLogic Server processes and participate in the standard server life cycle. Using WLDF, you can create, collect, analyze, archive, and access diagnostic data generated by a running server and the applications deployed within its containers. This data provides insight into the run-time performance of servers and applications and enables you to isolate and diagnose faults when they occur.

  Oracle Fusion Middleware Diagnostics Framework integrates with the following components of WLDF:

  – WLDF Policies and Actions, 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 Policies and Actions 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 What Is the WebLogic Diagnostics Framework? in *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*

- About Incidents and Problems
- Diagnostic Framework Components

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.
- Incident Flood Control

Incident Flood Control

It is conceivable that a problem could generate dozens or perhaps hundreds of incidents in a short period of time. This would generate too much diagnostic data, which would consume too much space in the ADR and could possibly slow down your efforts to diagnose and resolve the problem. For these reasons, the Diagnostic Framework applies flood control to incident generation after certain thresholds are reached. A **flood-controlled incident** is an incident that is not recorded in the ADR. Instead, the Diagnostic Framework writes a message at the WARNING level to the log file and returns an oracle.dfw.incident.Incident object. Flood-controlled incidents provide a way of informing you that a critical error is ongoing, without overloading the system with diagnostic data.

By default, if more than 5 incidents with the same problem key occur within 60 minutes, subsequent incidents with the same problem key are flood controlled. You can change this value using MBeans, as described in Configuring the Diagnostic Framework.

Diagnostic Framework Components

> **Note:**
>
> To use the Diagnostic Framework, in particular the Automatic Diagnostic Repository, the Managed Servers must have Oracle JRF applied. The following directory will exist for each Managed Server if Oracle JRF has been applied:
>
> *DOMAIN_HOME*/SERVERS/*server_name*/adr
>
> If the directory does not exist take one of the following steps:
>
> - Apply Oracle JRF, as described in Applying Oracle JRF Template to a Managed Server or Cluster.
> - If Oracle JRF has been applied, restart the servers, making sure that the Node Manager property `startScriptEnabled` is set to `true`, as described in Configuring Node Manager to Start Managed Servers.

The following topics describe the key components of the Diagnostic Framework:

- Automatic Diagnostic Repository
- Diagnostic Dumps
- Diagnostic Framework Management MBeans
- WLST Commands for Diagnostic Framework
- ADRCI Command-Line Utility

Automatic Diagnostic Repository

The Automatic Diagnostic Repository (ADR) is a file-based hierarchical repository for Oracle Fusion Middleware diagnostic data, such as traces and dumps. The Oracle Fusion Middleware components store all incident data in the ADR. Each Oracle WebLogic Server stores diagnostic data in subdirectories of its own home directory within the ADR. For example, each Managed Server and Administration Server has an ADR home directory.

The ADR root directory is known as ADR base. By default, the ADR base is located in the following directory:

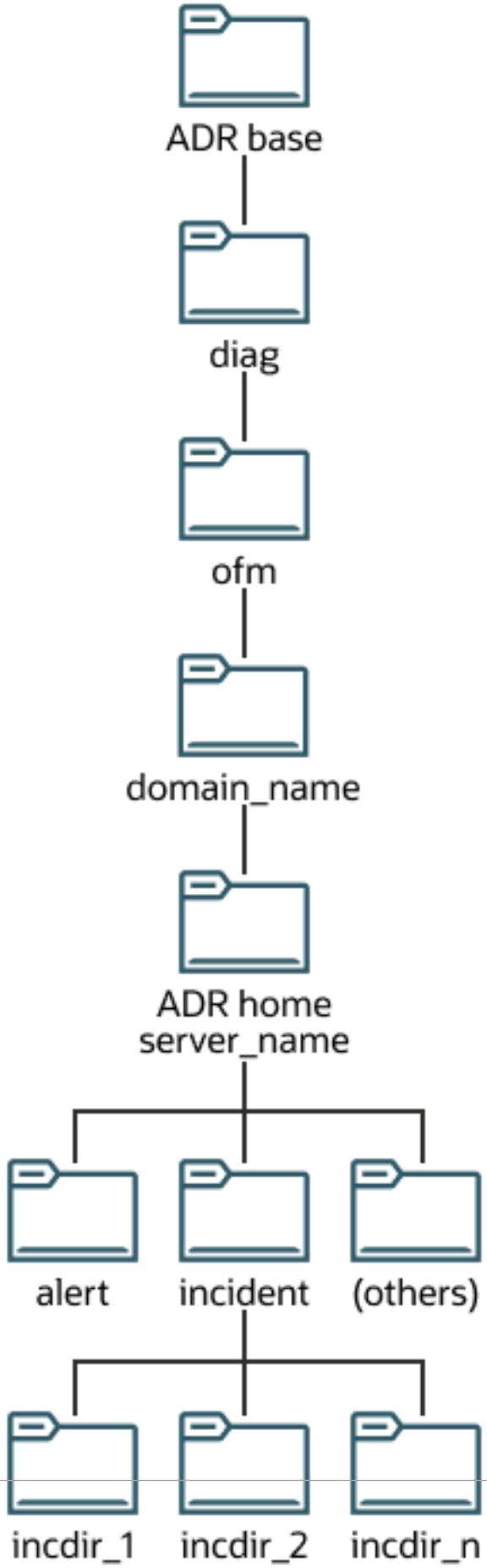
DOMAIN_HOME/servers/*server_name*/adr

Within ADR base, there can be multiple ADR homes, where each ADR home is the root directory for all incident data for a particular instance of Oracle WebLogic Server. The following path shows the location of the ADR home:

ADR_BASE/diag/ofm/*domain_name*/*server_name*

Figure 13-1 illustrates the directory hierarchy of the ADR home for an Oracle WebLogic Server instance.

Figure 13-1 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.

Diagnostic Dumps

A **diagnostic dump** captures and dumps specific diagnostic information when an incident is created (automatic) or on the request of an administrator (manual). When executed as part of incident creation, the dump is included with the set of incident diagnostics data. Examples of diagnostic dumps include a JVM thread dump, JVM class histogram dump, and DMS metric dump. For a list of diagnostic dumps, see #unique_362/unique_362_Connect_42_CHDICEEG.

Diagnostic Framework Management MBeans

The Diagnostic Framework provides MBeans that you can use to configure the Diagnostic Framework. For example, you can enable or disable flood control and you can configure how many incidents with the same problem key can occur within a specified time period. For information about using the management MBeans to configure the Diagnostic Framework, see Configuring the Diagnostic Framework.

You can also use the MBeans to query and create incidents, discover the list of available diagnostic dump types, and execute individual diagnostic dumps.

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:

- Viewing Problems
- Viewing Incidents
- Listing Diagnostic Dumps
- Viewing a Description of a Diagnostic Dump
- Executing Dumps
- Creating an Incident Manually
- Diagnostic Framework Custom WLST Commands in the *WLST Command Reference for Infrastructure Components*

ADRCI Command-Line Utility

The ADR Command Interpreter (ADRCI) is a utility that enables you to investigate problems, and package and upload first-failure diagnostic data to Oracle Support, all within a command-line environment. ADRCI also enables you to view the names of the dump files in the ADR, and to view the alert log with XML tags stripped, with and without content filtering.

ADRCI is installed in the following directory:

```
(UNIX) ORACLE_HOME/oracle_common/adr
(Windows) ORACLE_HOME\oracle_common\adr
```

See the following topics for information about using the ADRCI command-line utility:

- Packaging an Incident for information on packaging an incident.
- Purging Incidents for information on purging incidents.

See Also:

- ADRCI: ADR Command Interpreter in *Oracle Database Utilities*
- *Oracle Database Administrator's Guide*

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.

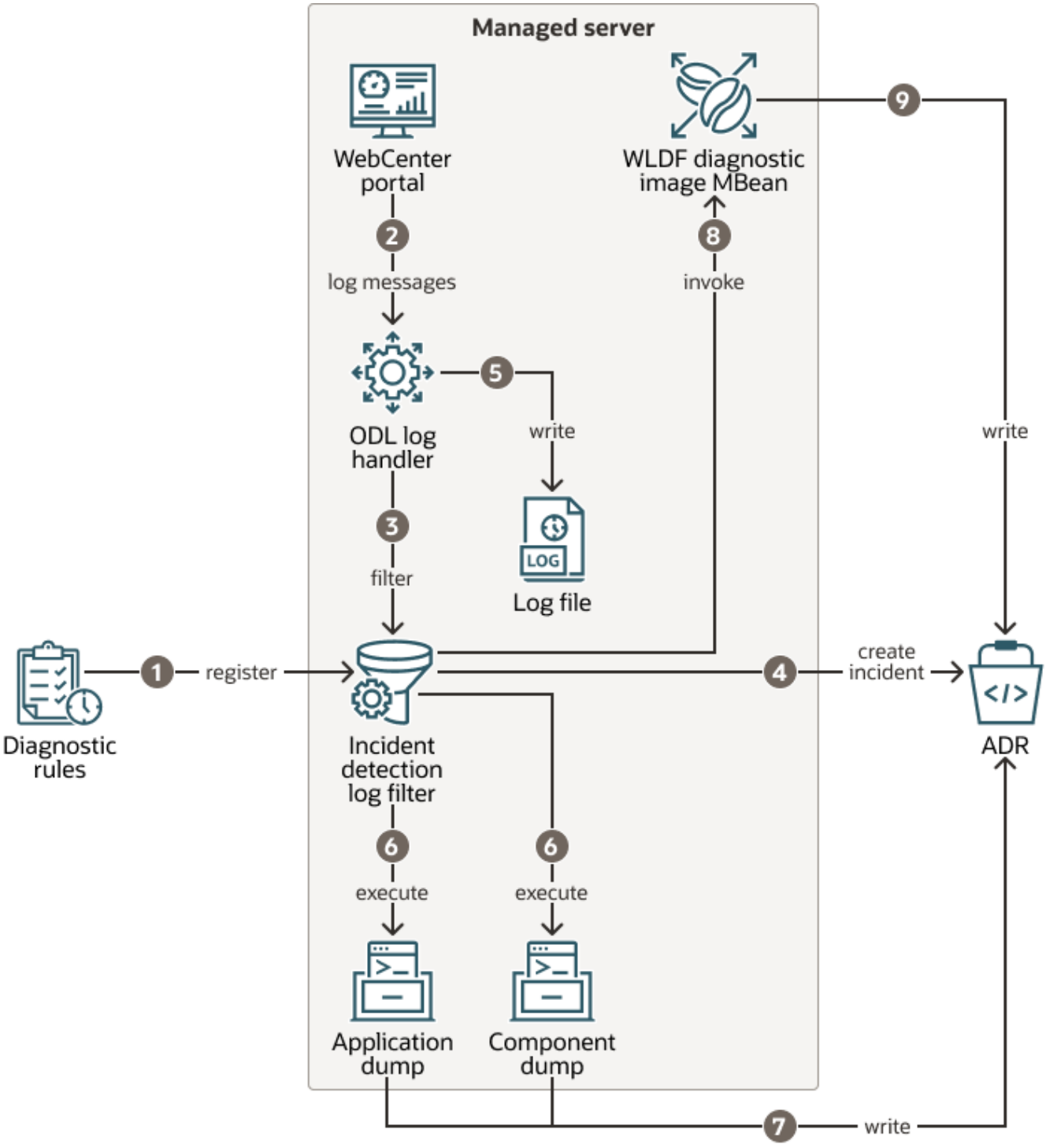
Incidents are automatically detected in two ways:

- By the incident detection log filter, which is automatically configured to detect critical errors.
- By the WLDF Policies and Actions component. The Diagnostics Framework listens for a predefined notification type and creates incidents when it receives such notifications.

  For information about configuring WLDF Policies and Actions, see Configuring WLDF Policies and Actions for the Diagnostic Framework.
- Programmatic incident creation. Some components create incidents directly.

Figure 13-2 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 13-2 Incident Creation Generated by Incident Log Detector



The steps represented in Figure 13-2 are:

1. The incident detection log filter is initialized with component and application diagnostic rules.
2. An application or component logs a message using the java.util.logging API.
3. The ODL log handler passes the message to the incident detection log filter.

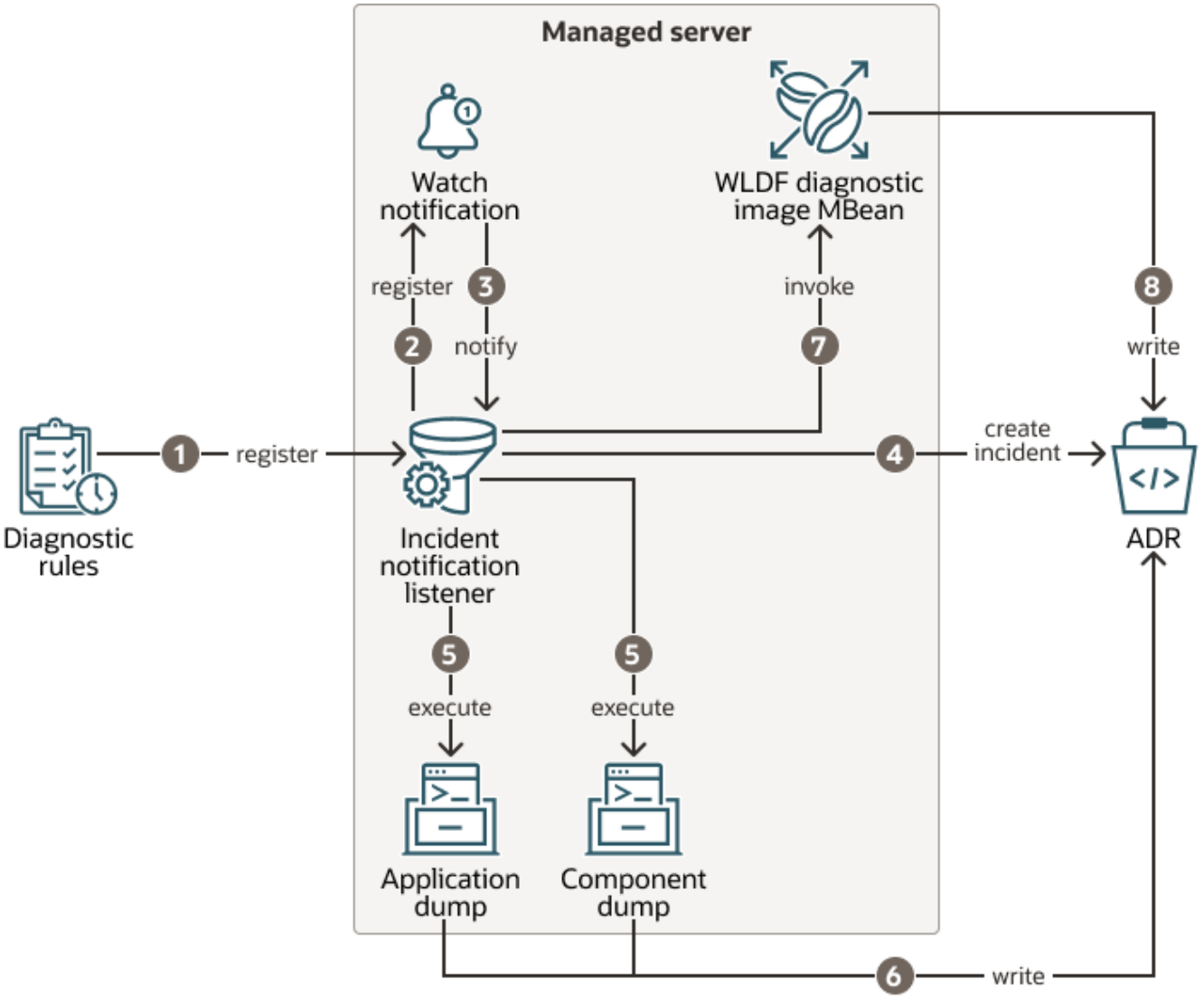
4. The incident log detection filter inspects the log message to see if an incident should be created, basing its decision on the diagnostic rules for the component. If the diagnostic rule indicates that an incident should be created, it creates an incident in the ADR.
5. The ODL log handler writes the log message to the log file, and returns control to the application.

   When an incident is created, a message, similar to the following, is written to the log file:

   ```
   [2017-03-28T11:05:34.603-07:00] [wls_server_1] [NOTIFICATION] [DFW-40101]
   [oracle.dfw.incident] [tid: [ACTIVE].ExecuteThread: '4' for queue:
   'weblogic.kernel.Default (self-tuning)'] [userId: weblogic] [ecid:
   66217af9-247f-4344-94a9-14f90e75a586-000e093f,0] An incident has been signalled with
   the incident facts: [problemKey=MDS-50500 [MANUAL] incidentSource=MANUAL
   incidentTime=Fri March 28 11:05:34 PDT 2017 errorMessage=MDS-50500
   executionContextId=null]
   ```
6. The Diagnostic Framework executes the diagnostic dumps that are indicated by the diagnostic rules for the component.
7. The Diagnostic Framework writes the dumps to ADR, in the directory created for the incident.
8. The Diagnostic Framework invokes the WLDF Diagnostic Image MBean requesting that a Diagnostic Image be created in ADR.
9. WLDF writes the Diagnostic Image to ADR.

Figure 13-3 shows the interaction when an incident is detected by the WLDF WLDF Policies and Actions system. It shows the interaction among the incident notification listener, the WLDF Policies and Actions system, and the WLDF Diagnostic Image MBean.

Figure 13-3 Incident Creation Generated by WLDF WLDF Policies and Actions



The steps represented in Figure 13-3 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 WLDF Policies and Actions system. It only processes notifications of type oracle.dfw.wldfnotification.
3. Something in the system causes the configured WLDF policy 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 policy 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.

Configuring the Diagnostic Framework

You can configure some settings for the Diagnostic Framework, including custom diagnostic rules and problem suppression. In addition, you can configure an WLDF Policies and Actions to create an incident.

The following topics describe how to configure the Diagnostic Framework:

- Configuring Diagnostic Framework Settings
- Configuring Custom Diagnostic Rules
- Configuring Problem Suppression
- Retrieving Problem Key Filters
- Configuring WLDF Policies and Actions for the Diagnostic Framework
  Oracle Fusion Middleware configures a WLDF Diagnostics Module that contains a set of Policy and Action rules (previously known as Watch and Notification rules) for detecting a specific set of critical errors and creating an incident for each occurrence of those errors. You can configure those Policies to create an incident.

Configuring Diagnostic Framework Settings

You can configure the following settings:

DiagnosticConfig MBean Attributes for Diagnostic Framework

- Enabling or disabling the detection of incidents through the log files
- Enabling or disabling flood control and setting parameters for flood control

You configure these settings by using the Diagnostic Framework MBean DiagnosticConfig. The following shows the MBean's ObjectName:

```
oracle.dfw:type=oracle.dfw.jmx.DiagnosticsConfigMBean,name=DiagnosticsConfig
```

#unique_369/unique_369_Connect_42_CHDDCHDI shows the attributes for the DiagnosticConfig MBean and a description of each parameter.

Attributes	Description
DumpSamplingIdleWhenHealthy	Determines whether dump sampling is active when the system is healthy. By default, this is set to true, which means that dump sampling is not active until an incident occurs.
DumpSamplingMinimumHealthyPeriod	The amount of time in seconds that the dump sampling is active after an incident occurs. The default is 259200 seconds (72 hours).
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.

Attributes	Description
floodControlIncidentCount	Sets the number of incidents with the same problem key that can be created within the time period, specified by floodControlIncidentTimeoutPeriod, before they are controlled by flood control. The default is 5. When flood control is enabled, if the number of incidents with the same problem key exceeds this count, no incidents are created, but the Diagnostic Framework writes a message at the WARNING level to the log file.
floodControlIncidentTimeoutPeriod	Sets the time period in which the number of incidents, as specified by floodControlIncidentCount, with the same problem key can be created before they are controlled by flood control. The default is 60 minutes.
incidentCreationEnabled	Enables or disables incident creation. 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`.
maxTotalIncidentSize	Sets the maximum total size that is allocated for all incidents. When the limit is reached, the oldest incidents are purged until the space used by all incidents is less than the amount specified by this parameter. The default is 500 MB. The limit may be exceeded during the creation of an incident, but when the incident creation completes, the oldest incidents are purged.
reservedMemoryKB	The amount of reserved memory that is released when OutOfMemoryError is detected. When the Diagnostic Framework starts, it allocates 512 KB of memory for its own private use. When the Diagnostic Framework detects that an OutOfMemoryError has occurred in the server, it frees that block of memory and proceeds to create the incident. The default is 512 KB.
uncaughtExceptionDetectionEnabled	Enables the Java-based uncaught exception handler. When enabled and an uncaught exception is detected, an incident is created. Specify `true` for enabled or `false` for disabled. The default is `true`.
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 WebLogic Domain menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.

2. Expand **Application Defined Beans**, then **oracle.dfw,** then **Domain.*domain_name,*** then **dfw.jmx.DiagnosticsConfigMBean.**

3. Select one of the **DiagnosticConfig** entries. There is one DiagnosticConfig entry for each server.

4. In the Application Defined MBean pane, expand **Show MBean Information** to see the server name.
5. To change the values for the attributes listed in #unique_369/unique_369_Connect_42_CHDDCHDI, enter or select the value in the **Value** field.
6. Click **Apply.**

Configuring Custom Diagnostic Rules

You can configure custom diagnostic rules that apply to a domain, a server, or an application in a domain or server.

You create the custom diagnostic rules by creating an .xml file with a particular format, which is shown in the example later in this section. You must save the file to one of the following locations:

Conditions for the LogDetectionConditions Element

- For rules that apply to the entire domain:

  DOMAIN_HOME/config/fmwconfig/dfw

- For rules that apply to a particular server:

  DOMAIN_HOME/config/fmwconfig/servers/*server_name*/dfw

The file name must use the following format:

name.xml
appname#*name*.xml

In the format, *appname* is the name of the application to which the rule applies. The *appname* must be the exact name of the deployed application. *name* is the name of the rule you specify. If you do not specify *appname*, the rules apply to the entire server. For example, the following rule applies to the application myApp:

```
myApp#custom_rule.xml
```

The custom diagnostic rules file can contain the following types of elements to define the rule:

- Log detection conditions, which are optional

  You can define a set of conditions, in the logDetectionConditions element, to check for in the diagnostic logs applicable to the server or to the specified application against which that the rules are registered. When a log message matching the condition is detected, an incident is created, capturing diagnostics that help identify the problem. By default, all INCIDENT_ERROR messages are detected and an incident created for them. In addition, specific components may have configured rules to detect specific messages.

  The following example shows a fragment of a custom diagnostic rules file that defines four log detection conditions. If one or more of the conditions are true, an incident is created.

```
<?xml version="1.0" encoding="UTF-8"?>
<diagnosticRules xmlns="http://www.oracle.com/DFW/DiagnosticsFrameworkRules"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
  <logDetectionConditions>
      <condition messageSeverity="INCIDENT_ERROR"/>
      <condition messageSeverity="ERROR" component="jrfServer_admin"/>
      <condition messageSeverity="ERROR" module="test.servletA"/>
      <condition messageId="FMW-40300"/>
    </logDetectionConditions>
```

See #unique_370/unique_370_Connect_42_BEIBEIEC for a description of the conditions you can use.

- Processing rules

  You can define processing rules that are evaluated when either the server or application rules are involved in incident creation. For example, if the application MyApp is involved in incident creation, any rules associated with the MyApp application are evaluated. In all cases, server-wide rules are evaluated regardless of the application.

  Processing rules consist of two parts:

  – Default actions, which are optional. If they are present, they are always executed during incident creation. The actions are a list of diagnostic dumps to execute, along with optional arguments.

    The following shows an example set of default actions:

    ```
    <defaultActions>
          <dumpAction name="odl.logs">
            <argument name="timestamp" value="INCIDENT_TIME" valueType="fact"/>
          </dumpAction>
          <dumpAction name="dms.metrics"/>
        </defaultActions>
    ```

    See #unique_370/unique_370_Connect_42_BEIFBHHJ for a description of the optional arguments that you can use.

  – Condition-based actions, which are executed only if the condition evaluates to true. Each <rule> element consists of a name attribute, along with a child <ruleCondition> element and a child <ruleActions> element. The <ruleActions> element contains one or more dumpAction elements. See #unique_370/unique_370_Connect_42_BEIFCDCH for a list of the <ruleCondition> element attributes.

    If multiple <condition> elements are specified in a single <rule> element, the dumpAction is executed only if all conditions evaluate to true.

    The following shows an example of a condition-based action rule. If the MESSAGE_ID is DFW-99997, the condition evaluates to true and the jvm.classhistogram dump is executed.

    ```
    <processingRules>
          <rule name="OOME">
            <ruleCondition>
              <condition name="MESSAGE_ID" value="DFW-99997"/>
            </ruleCondition>
            <ruleActions>
              <dumpAction name="jvm.classhistogram"/>
            </ruleActions>
          </rule>
        </processingRules>
    ```

#unique_370/unique_370_Connect_42_BEIBEIEC describes the attributes you can use to create the log detection conditions:

Condition	Description
messageSeverity	The log level at which the message was logged. (The MESSAGE_LEVEL field for ODL log files.) For example, INCIDENT_ERROR, ERROR.
messageId	The ID of the message. (The MESSAGE_ID field for ODL log files.) For example, DFW-99997.

Condition	Description
component	The component name. (The COMPONENT_ID field for ODL log files.) For example, `oracle.mds`.
module	The name of the module that originated the message. (The MODULE_ID field for ODL log files.)

See Table 12-1 for a description of the ODL log file fields.

#unique_370/unique_370_Connect_42_BEIFBHHJ describes the optional arguments that you can use for the <defaultActions> element.

Argument	Description
name	The name of the argument.
value	The value of the argument
type	The type of argument. Valid values are: • `literal`: If you specify this type, the literal value of the argument is used. This is the default. • `fact`: If you specify this type, the value must be either INCIDENT_TIME or ECID. • `context`: If you specify this type, the value must be the name of a value in the DMS Execution Context. For information on the DMS Execution Context, see DMS Execution Context in *Tuning Performance*.

#unique_370/unique_370_Connect_42_BEIFCDCH shows the <ruleCondition> element attributes.

Element	Description
name	The name of the attribute. Valid values depend on the valueType: • If the valueType is `fact`, valid values are COMPONENT_ID, MODULE_ID, or MESSAGE_ID. • If the valueType is `context`, the value must be the name of a value in the DMS Execution Context. For information on the DMS Execution Context, see DMS Execution Context in *Tuning Performance*.
operator	The operator. Value values are EQ, EQNoCase, NE, Contains, StartsWith, EndsWith, LT, GT, LE, GE. The default is EQ. The values are case sensitive.
value	The literal value to compare.
datatype	The data type. Valid values are String or Integer. The default is String. The values are case sensitive.
valueType	The type of argument: • `fact` • `context`

To create and load a custom diagnostic rule:

1. Create a file that contains the custom rules.

   The following shows a sample custom rules file:

```
<?xml version="1.0" encoding="UTF-8"?>
<diagnosticRules xmlns="http://www.oracle.com/DFW/DiagnosticsFrameworkRules"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">


    <logDetectionConditions>
      <condition messageSeverity="INCIDENT_ERROR"/>
           // detect all message logged at level INCIDENT_ERROR
      <condition messageSeverity="ERROR" component="jrfServer_admin"/>
          // detect all "jrfServer_admin" component messages logged at level ERROR
      <condition messageSeverity="ERROR" module="test.servletA"/>
          // detect all "test.servlet" module messages logged at level ERROR
      <condition messageId="FMW-40300"/>
          // detect message "FMW-40300"
    </logDetectionConditions>

    <defaultActions>
      <dumpAction name="odl.logs">
        <argument name="timestamp" value="INCIDENT_TIME" valueType="fact"/>
      </dumpAction>
      <dumpAction name="dms.metrics"/>
    </defaultActions>

    <processingRules>
      <rule name="OOME">
        <ruleCondition>
          <condition name="MESSAGE_ID" value="DFW-99997"/>
        </ruleCondition>
        <ruleActions>
          <dumpAction name="jvm.classhistogram"/>
        </ruleActions>
      </rule>
    </processingRules>

</diagnosticRules>
```

2. Save the file, naming it with the extension .xml. If the rule applies to an application, precede the file name with *app_name*#. Save the file to one of the following locations:

   DOMAIN_HOME/config/fmwconfig/dfw
   DOMAIN_HOME/config/fmwconfig/servers/*server_name*/dfw

3. Load the rules, using the WLST command reloadCustomRules. The following example loads the rule customrules.xml, which applies to the application myApp:

   ```
   reloadCustomRules(name='myApp#customrules.xml')
   ```

   You can reload all the rules in the domain or all the rules that pertain to a particular server. The following example reloads all the rules for the server wls_server1:

   ```
   reloadCustomRules(server='wls_server1')
   ```

   For more information about the reloadCustomRules command, see reloadCustomRules in the *WLST Command Reference for Infrastructure Components*.

Configuring Problem Suppression

In certain situations, you may want to suppress the creation of incidents based on a particular problem key. For example, in a development environment, when you are developing a servlet, you may generate high number of uncaught exceptions as you refine the code. This results in the creation of unnecessary incidents.

The Diagnostic Framework allows you to configure problem suppression filters so that problems that match the filter criteria do not result in the creation of an incident.

When you configure a problem suppression filter, you use a regular expression that represents a pattern that you want to match. The regular expression is matched using the java.util.regex class. For example:

DiagnosticConfig MBean Operations and Attributes for Problem Suppression Filters

- The following regular expression matches any incident with a problem key that starts with MDS-5000.

  ```
  MDS-5000.*
  ```

- The following regular expression matches any problem with the text `OutOfMemory`. Because the regular expression is case-sensitive, it will not match problems with the text `outofmemory`.

  ```
  .*OutOfMemory.*
  ```

You can add and remove filters and get a list of filters or the detail of one filter using the DiagnosticConfig MBean.

#unique_371/unique_371_Connect_42_BEIFADJA shows the operations and attribute for configuring problem suppression filters and a description of each.

Operations and Attribute	Description
Operation: addProblemKeyFilter(*filter_pattern*)	Adds a new problem suppression filter. You pass it the regular expression that represents a pattern that you want to match. For example: `addProblemKeyFilter(".*OutOfMemory.*)`
Attribute: getProblemKeyFilters()	Returns a list of the configured problem suppression filters. For example: `getProblemKeyFilters()`
Operation: getProblemKeyFilter(*filterID*)	Returns the filter pattern associated with the specified ID. For example: `getProblemKeyFilter(id)` To find the ID, use the getProblemKeyFilters() operation.
Operation: removeProblemKeyFilter(*filterID)*	Removes the filter pattern associated with the given filter ID. For example: `removeProblemKeyFilter(id)`

To configure a problem suppression filter:

1. From the WebLogic Domain menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.

2. Expand **Application Defined Beans**, then **oracle.dfw,** then **domain.*domain_name,*** then **dfw.jmx.DiagnosticsConfigMBean.**

3. Select one of the **DiagnosticConfig** entries. There is one DiagnosticConfig entry for each server.

4. In the Application Defined MBeans pane, select the Operations tab.
5. Click **addProblemKeyFilter.** The Operation: addProblemKeyFilter page is displayed.
6. For **Value,** enter a regular expression that represents a pattern that you want to match pattern. For example, in a development environment, you might want to add a filter so that incidents are not created when java.lang.IllegalStateException Java Exceptions are reported. In that case, enter the following:

   ```
   ".*[java.lang.IllegalStateException].*"
   ```

7. Click **Invoke.**
8. Click **Return** to return to the Application Defined MBeans page.

You can delete the filters using the removeProblemKeyFilter operation.

Retrieving Problem Key Filters

You can retrieve a specific filter, passing the ID of the filter to the getProblemKeyFilter operation.

Alternatively, you can retrieve a list of the filters using the getProblemKeyFilters attribute:

1. From the WebLogic Domain menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.
2. Expand **Application Defined Beans**, then **oracle.dfw,** then **domain.*domain_name,*** then **dfw.jmx.DiagnosticsConfigMBean.**
3. Select one of the **DiagnosticConfig** entries. There is one DiagnosticConfig entry for each server.
4. In the Application Defined MBeans pane, select the Attributes tab.
5. Click **ProblemKeyFilters.**

   The list of problem suppression filters is displayed.

Configuring WLDF Policies and Actions for the Diagnostic Framework

Oracle Fusion Middleware configures a WLDF Diagnostics Module that contains a set of Policy and Action rules (previously known as Watch and Notification rules) for detecting a specific set of critical errors and creating an incident for each occurrence of those errors. You can configure those Policies to create an incident.

The WLDF Diagnostics Module is called Module-FMWDFW and contains the following set of Policy 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 Policy conditions to create an incident:

1. In Fusion Middleware Control, from the WebLogic Domain menu, expand Diagnostics and select **Diagnostic Modules.**

   The Diagnostic Modules page is displayed.

2. Click **Module-FMWDFW.**

   The Module-FMWDFW page is displayed.

3. Select the Configuration tab, then the Policies and Actions tab.

4. Select the Policies tab and click **Create.**

   The Create a Diagnostic Policy page is displayed.

5. For **Policy Name,** enter a name for the policy.

   You can enter any name. Alternatively, you can use the following format to force the Diagnostic Framework to use a custom message ID:

   message-id#[*application_name*]#*any_text*

   The message ID consists of a prefix that can be 1 to 6 characters, and a number, that can be 1 to 6 digits. The application name is optional. For example:

   ```
   WLS-40500#My_Policy_Name
   ```

   The following example uses the application name testapp:

   ```
   WLS-40501#testapp#My_Policy_Name
   ```

   The Diagnostic Framework uses the message ID as the incident message ID in constructing the incident problem key.

6. For **Rule Type,** select a type, for example, Server Log or Smart Rule based, which are pre-configured rules, such as Cluster Low Average Throughput.

7. Click **Next.**

8. The next pages depend on the Policy type:

   - For Smart Rule based:

     a. Select a rule and click **Next.**

     b. Provide values for the parameters and click **Next.**

     c. If you intend the policy to be scheduled on particular days of the week or month, for **Start Time,** provide values for **Hour, Minute,** and **Second.** Then, select **AM** or **PM.**

        This option is used only when you select **Specific days of the week** or **Specific days of the month** from the **Repeat** field.

     d. For **Repeat**, select the number of times it will be run within a specified time. For example, select **Every N minutes**.

     e. If you intend the policy to be scheduled to be run at specified intervals, such as every five minutes, for **Frequency**, enter a value, such a 5. If you selected Every N minutes for Repeat, then the schedule would be every 5 minutes. Click **Next.**

     f. Select an alarm type and click **NEXT.**

     g. For Scaling Actions, select either **Scale Up Action** or **Scale Down Action.**

h. For **Diagnostic Actions,** move an action from the **Available** column to the **Chosen** column.

- For Calendar Based:
  a. If you intend the policy to be scheduled on particular days of the week or month, for **Start Time,** provide values for **Hour, Minute,** and **Second.** Then, select **AM** or **PM.**

     This option is used only when you select **Specific days of the week** or **Specific days of the month** from the **Repeat** field.

  b. For **Repeat**, select the number of times it will be run within a specified time. For example, select **Every N minutes**.

  c. If you intend the policy to be scheduled to be run at specified intervals, such as every five minutes, for **Frequency**, enter a value, such a 5. If you selected Every N minutes for Repeat, then the schedule would be every 5 minutes. Click **Next.**

  d. If you have a dynamic cluster, for **Scaling Actions,** select either **Scale Up Action** or **Scale Down Action.**

  e. For **Diagnostic Actions,** move an action from the **Available** column to the **Chosen** column.

- For Collected Metrics:
  a. Select **Smart Rule** or **Expression.** Click **Next.**

     If you select Smart Rule, provide values for the parameters.

     If you select Expression, enter an expression.

  b. Click **Next.**

  c. If you intend the policy to be scheduled on particular days of the week or month, for **Start Time,** provide values for **Hour, Minute,** and **Second.** Then, select **AM** or **PM.**

     This option is used only when you select **Specific days of the week** or **Specific days of the month** from the **Repeat** field.

  d. For **Repeat**, select the number of times it will be run within a specified time. For example, select **Every N minutes**.

  e. If you intend the policy to be scheduled to be run at specified intervals, such as every five minutes, for **Frequency**, enter a value, such a 5. If you selected Every N minutes for Repeat, then the schedule would be every 5 minutes. Click **Next.**

  f. Select an alarm type and click **NEXT.**

  g. For Scaling Actions, select either **Scale Up Action** or **Scale Down Action.**

  h. For **Diagnostic Actions,** move an action from the **Available** column to the **Chosen** column.

- For Domain Log:
  a. Add expressions to create the rule for your policy. Then, click **Next.**

  b. Select an alarm type and click **NEXT.**

  c. If you have a dynamic cluster, for **Scaling Actions,** select either **Scale Up Action** or **Scale Down Action.**

  d. For **Diagnostic Actions,** move an action from the **Available** column to the **Chosen** column.

- For Event Data:

a. Add expressions to create the rule for your policy. Then, click **Next.**

b. Select an alarm type and click **NEXT.**

c. If you have a dynamic cluster, for **Scaling Actions,** select either **Scale Up Action** or **Scale Down Action.**

d. For **Diagnostic Actions,** move an action from the **Available** column to the **Chosen** column.

- For Server Log:

  a. Add expressions to create the rule for your policy. For example, you can construct the expression (SEVERITY = 'Error') and (MSGID = 'BEA-000337'). Then, click **Next.**

  b. Select an alarm type and click **NEXT.**

  c. If you have a dynamic cluster, for **Scaling Actions,** select either **Scale Up Action** or **Scale Down Action.**

  d. For **Diagnostic Actions,** move an action from the **Available** column to the **Chosen** column.

9. Click **Create.**

Investigating, Reporting, and Solving a Problem

You can use WLST and ADRCI commands and the Remote Diagnostic Agent (RDA) to investigate and report a problem (critical error), and in some cases, resolve the problem.

The section begins with a roadmap that summarizes the typical set of tasks that you must perform. It describes the following topics:

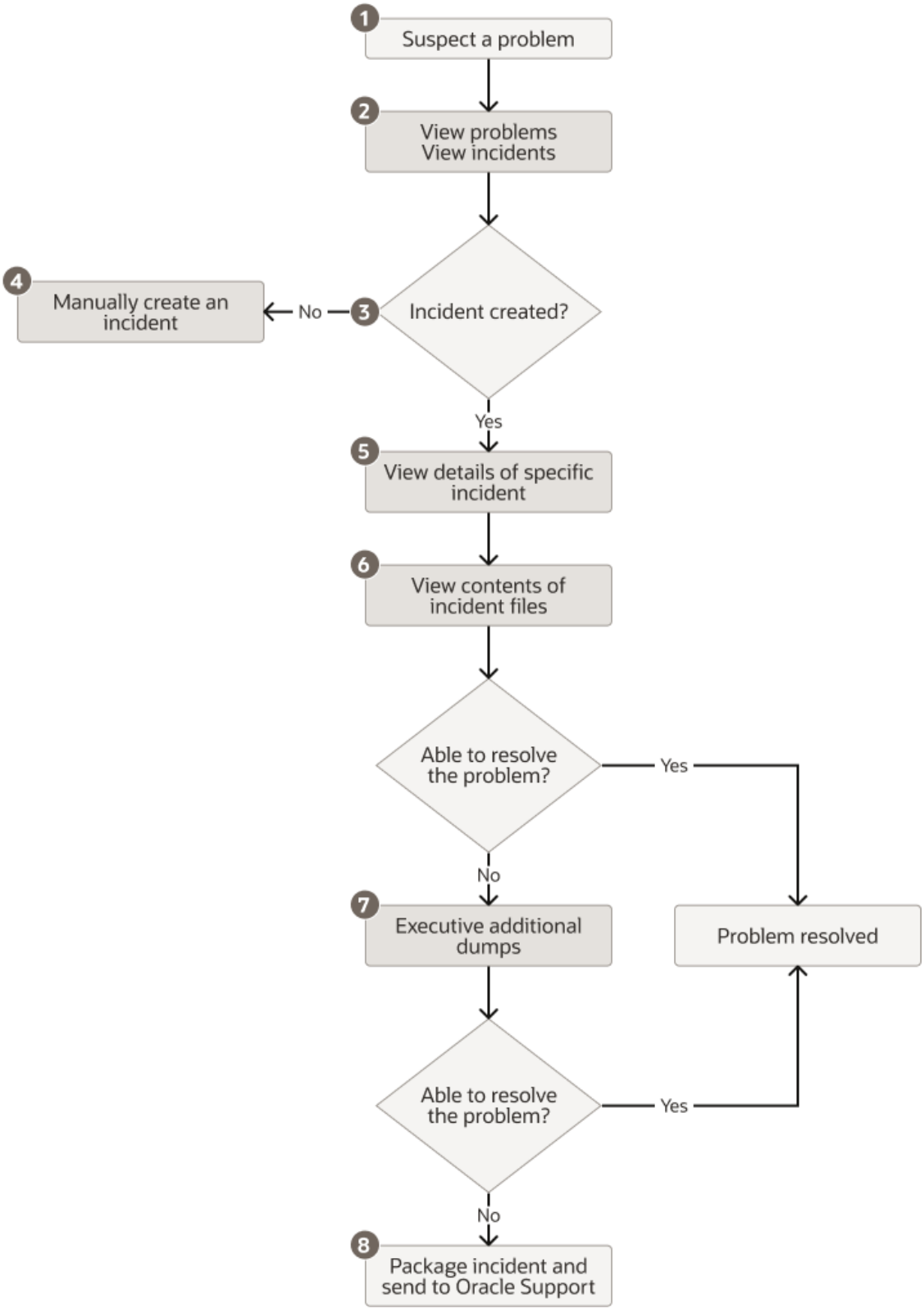
- Roadmap—Investigating, Reporting, and Resolving a Problem
- Viewing Problems and Incidents
- Analyzing Specific Problem Keys
- Working with Diagnostic Dumps
- Configuring and Using Diagnostic Dump Sampling
- Managing Incidents
- Generating a RDA Report

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 13-4 illustrates the tasks that you complete to investigate, report, and resolve a problem.

Figure 13-4 Flow for Investigating a Problem



The following describes the workflow illustrated in Figure 13-4:

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 Viewing Problems.
   b. If a problem has been created, list the incidents related to the specific problem using the `listIncidents` command, as described in Viewing Incidents.
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 Creating an Incident Manually for more information about creating an incident.
5. View the details of the specific incident using the `showIncident` command, as described in Viewing Incidents. 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 Viewing Incidents. 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 Working with Diagnostic Dumps for information about the dumps available and how to execute them.
8. If you still cannot resolve the problem, package the incident, along with the RDA report, and send them to Oracle Support. See Packaging an Incident and Generating a RDA Report for information about packaging incidents and generating RDA reports.

Viewing Problems and Incidents

You can view the set of problems, the list of incidents, and the details of a particular incident using the WLST command-line utility, as described in the following topics:

- Viewing Problems
- Viewing Incidents
- Querying Incidents

Viewing Problems

You can view the set of problems by executing the WLST `listProblems` command, using the following format:

```
listProblems([adrHome] [,server])
```

The `listProblems` command lists the problems in the ADR home. Each problem has a unique ID:

```
listProblems()
Problem Id      Problem Key
        1       BEA-101020 [HTTP]
```

Viewing Incidents

You can list of all available incidents or the incidents related to a specific problem by executing the WLST `listIncidents` command, using the following format:

```
listIncidents([id], [ADRHome])
```

For example, to see the list of all incidents, use the following command:

```
listIncidents()
Incident Id     Incident Time                   Problem Key
        2       Fri Apr 28 11:05:59 PDT 2017    MDS-50500 [MANUAL]
        1       Fri Apr 28 11:02:22 PDT 2017    MDS-50500 [MANUAL]
```

To view the incidents related to a specific problem, use the following command:

```
listIncidents(id='1')
Incident Id     Incident Time                   Problem Key
        2       Fri Apr 28 11:05:59 PDT 2017    MDS-50500 [MANUAL]
        1       Fri Apr 28 11:02:22 PDT 2017    MDS-50500 [MANUAL]
```

To view the details of a particular incident, use the WLST `showIncident` command, using the following format:

```
showIncident(id, [adrHome] [,server])
```

For example, to see the details of incident 1, use the following command:

```
showIncident(id='1')
Incident Id: 1
Problem Id: 1
Problem Key: MDS-50500 [MANUAL]
Incident Time:Fri Apr 28 11:02:22 PDT 2017
Error Message Id: MDS-50500
Execution Context:
Flood Controlled: false
Dump Files :
   readme.txt
   jvm_threads10_i1.txt
   dms_metrics11_i1.txt
   dfw_samplingArchive13_i1.JVMThreadDump.txt
   dfw_samplingArchive13_i1.readme.txt
   odl_logs14_i1.txt
```

To view the contents of a file in the incident, use the WLST `getIncidentFile` command, using the following format:

```
getIncidentFile(id, name [,outputFile] [,adrHome] [,server])
```

For example, to view the contents for the file odl_logs4_i1.dmp use the following command:

```
getIncidentFile(id='1', name='odl_logs14_i1.txt',outputFile='/tmp/
odl_logs4_i1_dmp.output')
```

The command writes the output to the file odl_logs4_i1_dmp.output.

Querying Incidents

While the `listIncidents` command shows you the incidents related to a particular problem ID, or for a particular server, it does not allow you to restrict the list further. The WLST `queryIncidents` command lets you query for the value of particular attributes across one or more servers, or all servers in a domain. For example, you can query by the time of incident creation or the ECID.

An expression contains an incident attribute, an operator, and a string, in the following format:

```
attribute operator "string"
```

You can combine query expressions with the Boolean operators AND or OR, and group them by parentheses ().

The following incident attributes are supported:

- TIMESTAMP: Incident creation time. You can use the `from` and `to` operators to specify a time range. The date format is `YYYY-MM-DD HH:MM`.
- ECID: Execution Context ID
- PROBLEM_KEY: Problem Key
- MSG_FACILITY: The error message facility, such as ORA or OHS.
- MSG_NUMBER: The error message ID, such as 600.

Custom incident attributes are also supported. For example, TRACEID, APP, URI, and DSID are supported. In addition, the context values, as shown in the incident readme.txt file, are supported. For example, DFW_APP_NAME and DFW_USER_NAME are supported.

The following operators are supported:

- equals
- notEqual
- startsWith
- endsWith
- contains
- isNull
- notNull

For example, you can query all incidents in all servers in the domain for the ECID f19wAgN000001:

```
queryIncidents(query="ECID equals f19wAgN000001")
```

The following example queries all incidents that occurred between March 1, 2017 and March 15, 2017, for the server wls_server_1:

```
 queryIncidents(query="TIMESTAMP from '2017-03-01 00:00'AND TIMESTAMP to '2017-03-15
00:00'",
servers=["wls_server_1"])
```

For the complete syntax for this command, see queryIncidents in the *WLST Command Reference for Infrastructure Components*.

Analyzing Specific Problem Keys

Uncaught Exception Problem Keys

The Diagnostic Framework provides a set of well-defined problem keys for unhandled exceptions. These exceptions are either detected through the existing WLDF Policy "UncheckedException" or through the Diagnostic Framework java.lang.Thread.UncaughtExceptionHandler handler. Previously, the Diagnostic Framework generated problem keys with different formats for the same type of issues. #unique_375/unique_375_Connect_42_BEIEGJEE describes these problem keys and how to use them to investigate a problem.

Exception	Problem Key	Description
java.lang.OutOfMemoryError	DFW-99997 [java.lang.OutOfMemoryError]	Used by all java.lang.OUtOfMemoryError incidents. With each incident of this type, a jvm.classhistogram dump is executed. The dump captures statistics about the instances of classes that have been loaded and the counts of associated Objects. Review the contents of this dump for a good starting point for understanding what has been loaded into the JVM's memory. In addition, the dms.metrics dump records statistics about the overall JVM memory.
java.sql.SQLException	DFW-99996 [*ora-code*| java.sql.SQLException]] [*package.class.method*][*app-name*]	Used for all exceptions of type java.sql.SQLException, including its subclasses. The Diagnostic Framework attempts to extract the Oracle error code from the exception error message, and if it is successful, uses that in the problem key. If not, it uses the exception name. Review the text associated with the exception to get more details, such as the operation that could not be performed on the database. In addition, you can review the SQL error code details for additional information.
All others	DFW-99998 [*exception-name*] [*package.class.name*][*app-name*]	Used by all other types of exceptions, such as java.lang.NullPointerException, java.io.IOException, java.lang.StringIndexOutOfBoundsException, that are not handled in a unique way. Review the text associated with the exception to get more details, such as the reason for the failure. The source line in the problem key is a best-attempt indicator of the location of the failure.

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 Jakarta 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

Listing Diagnostic Dumps

Diagnostic Dump Actions

You can find a list of diagnostic dumps that are available for a Managed Server by executing the WLST `listDumps` command, using the following format:

```
listDumps([appName] [,server])
```

For example, to list the available dumps for wls_server1:

```
listDumps(server='wls_server1')
Location changed to domainRuntime tree. This is a read-only tree with DomainMBean as the
root.
For more help, use help(domainRuntime)
dfw.samplingArchive
dms.configuration
dms.ecidctx
dms.metrics
http.requests
jvm.classhistogram
jvm.threads
mds.MDSInstancesDump
odl.activeLogConfig
odl.logs
odl.quicktrace
opss.diagTest
opss.identityStoreUserRoleApiConfig
opss.securityContext
wls.image

Use the command describeDump(name=<dumpName>) for help on a specific dump.
```

#unique_362/unique_362_Connect_42_CHDICEEG lists the diagnostic dump actions that are defined by Oracle Fusion Middleware and their descriptions.

Dump Action	Description
dms.ecidctx	The data associated with a specific Execution Context ID (ECID), if specified. Otherwise, the data associated with all available ECIDs.
dms.metrics	Dynamic Monitoring Service (DMS) metrics. For information about these metrics, see About Dynamic Monitoring Service (DMS) in *Tuning Performance.*
http.requests	A summary of the currently active HTTP requests.

Dump Action	Description
jvm.classhistogram	A JVM class histogram, the output of which varies depending on the JVM vendor.
jvm.flightRecording	The active JRockit Flight Recorder recording.
jvm.threads	Summary statistics about the threads running in a JVM as well as performing a full thread dump.
mds.MDSInstancesDump	Information about each MDS instance in the current JVM.
odl.activeLogConfig	The active Java logging configuration.
odl.logs	Contents of diagnostic logs, correlated by ECID or time range.
odl.quicktrace	Quick trace messages.
wls.image	The WLDF server image dump.

In addition, Oracle SOA Suite provides diagnostic dumps, as described in Diagnosing Problems with SOA Composite Applications in *Administering Oracle SOA Suite and Oracle Business Process Management Suite.*

Viewing a Description of a Diagnostic Dump

You can view a description of a particular dump, including the syntax for executing the dump by using the WLST `describeDump` command. You specify the name of the dump in which you are interested. For example, to view a description of the dms.metrics dump, use the following command:

```
describeDump(name='dms.metrics')
Name: dms.metrics
Description: Dumps DMS (Dynamic Monitoring Service) metrics.
Run Mode: asynchronous
Mandatory Arguments:
Optional Arguments:
    Name        Type     Description
    dump        STRING   How much to dump
    servers     STRING   Server names
    names       STRING   Name of DMS noun or metric
    format      STRING   Format of the dump output; raw or xml
    nountypes   STRING   Type of DMS noun
```

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 Viewing a Description of a Diagnostic Dump.

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 command writes the dump output to the information about incident 1. If you execute the showIncident command for incident 1, the output includes dms_metrics1_i1.dmp.

Configuring and Using Diagnostic Dump Sampling

Diagnostic dump sampling captures the output of diagnostic dumps at specified intervals. By sampling at regular intervals, diagnostic dump sampling can help to reveal issues such as slow running web requests, and where work is being performed in those requests.

This section contains the following topics:

About Diagnostic Dump Sampling

All diagnostic dump samplings are performed in the background, at specified intervals. By default, jvm.threads and jvm.classhistogram dumps are configured for sampling. However, they are not active until an incident is generated. Then, they remain active for 72 hours, by default.

You can modify the settings for the default dump samplings and you can create new sampling definitions for the dump actions listed in #unique_362/unique_362_Connect_42_CHDICEEG and for any application-specific dumps. You can configure multiple sampling definitions for the same diagnostic dump, specifying different settings, such as sampling interval or server.

For each diagnostic dump sampling, the Diagnostic Framework stores the specified number of samples. When that limit is reached, the oldest sample is purged. All samples are purged when the server shuts down.

Table 13-1 shows the settings of the dump samplings that are configured by default.

Table 13-1 Default Diagnostic Dump Samplings Configuration

Dump Name	Sampling Interval	Maximum Samples Stored
jvm.threads	60 seconds	10
jvm.classhistogram	30 minutes	5

The Diagnostic Framework triggers the retrieval of the dump samples whenever an incident is created (through error detection or manual incident creation.) In addition, you can retrieve the contents of the dump samples, as described in Retrieving the Dump Sampling Output.

You can retrieve the dump sample archives in either text or zip files:

- Text: By default, the diagnostic dump samples are concatenated into a single archive file, in text format. An ASCII header an footer are wrapped around each sample in the archive file. The header contains a timestamp and the name of the diagnostic dump that produced the sample. Both the header and footer contain the number of the samples in the archive and the number of the particular sample. For example:

```
$$$=== BEGIN OF Diagnostic Dump - jvm.classhistogram (Archive #0 1_of_2) ===$$$
Fri Apr 28:00:00 PDT 2017

<text of dump sampling>

$$$=== END OF Diagnostic Dump - jvm.classhistogram (Archive #0 1_of_2) ===$$$
```

- Zip: You can configure diagnostic dump samplings to return a zip file instead of a concatenated file. The zip file contains all available dump sample files. This format supports any diagnostic dumps whose outputs are in binary format not suitable for concatenation, as well as for dumps that generate output in text format. This format also reduces the size of the archive containing the samples.

  The following example shows the contents of a zip file:

```
unzip -l jvm_dump.zip
Archive:  jvm_dump.zip
  Length      Date    Time    Name
 --------    ----    ----    ----
   508780  04-28-17 07:25   dfw_samplingArchive1065570966467923683.JVMThreadDump.dmp
      840  04-28-17 07:25   dfw_samplingArchive7749640004639161119.readme.txt
 --------                   -------
   509620                   2 files
```

In addition to a text or zip file, when you retrieve a dump sample, the Diagnostic Framework generates a readme file. The readme file either lists the line numbers for each dump sample in the archive (for text format) or the individual sample file names (for zip format). It also lists the timestamp for each sample and the index for the archive.

The dump sample files are named using the following format:

`dfw_dumpArchive`*`nnn`*`.`*`Sampling_Name`*`.{txt]|zip}`

In the format *`nnn`* is a unique number assigned by the Diagnostic Framework.

For example, the following is an example of the name of a dump sample file for JVMThreadDump:

```
dfw_dumpArchive17394218037.JVMThreadDump.txt
```

The readme files are named using the following format:

`dfw_dumpArchive`*`nnn`*`.readme.txt`

In the format *`nnn`* is a unique number assigned by the Diagnostic Framework.

All samplings are scheduled to begin at the next nearest interval, corresponding to the frequency. For example, if a sampling is configured at 12:05:13 PM and the frequency is 5 seconds, the sample will be collected at 12:05:15 PM. This ensures that the collection of a series of samplings with the same frequency will occur at the same time. It also aligns all samples across machines, assuming their system clocks are synchronized.

> **Note:**
>
> You must be connected to the Administration Server to execute the WLST dump sampling commands.

Configuring Dump Sampling

You can create additional dump samplings, update existing dump samplings, remove dump samplings and enable or disable dump sampling, as described in the following topics:

- Activating the Default Samples
- Creating Dump Samplings

- Modifying Dump Sampling Settings
- Removing Dump Samplings
- Enabling or Disabling All Dump Sampling

Activating the Default Samples

By default the jvm.threads and jvm.classhistogram dumps are not activated until an incident occurs. Then, they are active for 72 hours, by default.

You can change the behavior so that the dumps are active even if an incident has not occurred by setting the value of the MBean DumpSamplingIdleWhenHealthy to `false`.

To change the amount of time used for determining the system's health, change the value of the DumpSamplingMinimumHealthyPeriod MBean.

For information about changing the value of the Diagnostic Framework MBeans, see Configuring Diagnostic Framework Settings

Creating Dump Samplings

You can create dump samplings for any dump listed in #unique_362/unique_362_Connect_42_CHDICEEG and for any application-specific dumps. To create dump samplings, use the WLST command addDumpSample. The addDumpSample command uses the following syntax:

```
addDumpSample(sampleName="sample_name", diagnosticDumpName="dump_name",
 [appName="application_name",] samplingInterval=num_seconds,
 rotationCount=num_samples, [dumpedImplicitly={true|false},]
 [toAppend={true|false},] [args={"arg_name" : "value"},]
 [server="server_name"])
```

For example, to create a dump sampling for the http.requests dump, setting the sampling interval to 300 seconds and the rotation count to 10 samples, for the server wls_server1:

```
addDumpSample(sampleName="HTTPSampling", diagnosticDumpName="http.requests",
samplingInterval=300, rotationCount=10, server="wls_server1")

HTTPSampling is added
```

For complete syntax, see addDumpSample in the *WLST Command Reference for Infrastructure Components.*

Modifying Dump Sampling Settings

You can change the settings of existing dump samplings by using the WLST command updateDumpSample. The updateDumpSample command uses the following syntax:

```
updateDumpSample(sampleName="sample_name",
        [appName="application_name",] samplingInterval=num_seconds,
         rotationCount=num_samplings, [dumpedImplicitly={true|false},]
        [toAppend={true|false},] [args={"arg_name" : "value"},]
        [server="server_name"])
```

For example, to modify the dump sampling HTTPSampling, changing the sampling interval to 200 and the rotation count to 5:

```
updateDumpSample(sampleName="HTTPSampling", samplingInterval=200,
                  rotationCount=5, server="wls_server1")
```

```
HTTPSampling is updated
```

For complete syntax, see updateDumpSample in the *WLST Command Reference for Infrastructure Components.*

Removing Dump Samplings

You can remove existing dump samplings using the WLST command removeDumpSample. The removeDumpSample command uses the following syntax:

```
removeDumpSample(sampleName="sample_name", [server="server_name"])
```

For example, to remove the dump sampling HTTPSampling:

```
removeDumpSample(sampleName="HTTPSampling", server="wls_server1")

Removed HTTPSampling
```

For complete syntax, see removeDumpSample in the *WLST Command Reference for Infrastructure Components*.

Enabling or Disabling All Dump Sampling

You can enable or disable all dump sampling using the WLST command enableDumpSampling. This command affects all configured dump samplings. The enableDumpSampling command uses the following syntax:

```
enableDumpSampling(enable={true|false}, [server="server_name"])
```

Note that the server parameter is valid only if you are connected to the Administration Server. If you do not specify the server parameter, dump sampling is disabled for the Administration Server.

For example, to disable dump sampling for the Administration Server:

```
enableDumpSampling(enable=false)

Dump sampling disabled
```

To determine if dump sampling is enabled or disabled, use the WLST command isDumpSamplingEnabled. The isDumpSamplingEnabled command uses the following format:

```
isDumpSamplingEnabled([server="server_name"])
```

For complete syntax, see enableDumpSampling and isDumpSamplingEnabled in the *WLST Command Reference for Infrastructure Components*.

Listing Dump Samplings

You can list dump samplings using the WLST command listDumpSamples. You can list all dump samplings, a specified dump sampling, or all dump samplings associated with a specified server. The listDumpSamples command uses the following syntax:

```
listDumpSample([sampleName="sample_name",] [server="server_name"])
```

For example, to list all dump samplings associated with the server wls_server1:

```
listDumpSamples(server="wls_server1")
Name              : JVMThreadDump
```

```
Dump Name         : jvm.threads
Application Name  :
Sampling Interval : 30
Rotation Count    : 20
Dump Implicitly   : true
Append Samples    : true
Dump Arguments    : context=true, timing=true, progressive=true, depth=20,
threshold=30000

Name              : JavaClassHistogram
Dump Name         : jvm.classhistogram
Application Name  :
Sampling Interval : 1800
Rotation Count    : 5
Dump Implicitly   : false
Append Samples    : true
Dump Arguments    :
```

For complete syntax, see listDumpSample in the *WLST Command Reference for Infrastructure Components.*

Retrieving the Dump Sampling Output

To retrieve the output of dump samples, you can use the WLST executeDump command or the WLST getSamplingArchives command, as described in the following topics:

- Retrieving Dump Samples Using the executeDump Command
- Retrieving Dump Samples Using the getSamplingArchives Command

Retrieving Dump Samples Using the executeDump Command

You can retrieve dump samples using the WLST executeDump command, specifying the dfw.samplingArchive dump. This command collects all default sample archives and any dump samples that are specified with the parameter `dumpImplicitly=true` from a temporary location and concatenates them into a single file. The command also returns a readme file, with details of the dump samples.

When you use the executeDump command, you use the following syntax:

```
executeDump(name="dfw.samplingArchive",outputFile="filename"
```

For the outputFile parameter, you can specify a text file or a zip file. If you specify a zip file, you must use the argument `zipOutput=true`.

For any dump sampling that is configured with the parameter `dumpImplicitly=false`, you must specify the optional dfw.samplingArchive argument `sampleName` to collect the contents of those dump samples. For example:

```
executeDump(name='dfw.samplingArchive', args={'sampleName' : 'JavaClassHistogram'})
```

For complete syntax for this command, see executeDump in the *WLST Command Reference for Infrastructure Components*.

Retrieving Dump Samples Using the getSamplingArchives Command

You can retrieve dump samples using the WLST getSamplingArchives command. This command collects all dump samples in a zip file containing the individual dump sample files and a readme file. This method is particularly useful in dealing with binary format dumps.

The getSamplingArchives command uses the following syntax:

```
getSamplingArchives([sampleName="sample_name"] [,outputFile="filename"
[,server="server_name"])
```

For example to retrieve the dump samples for the sampling JavaClassHistogram, use the following command:

```
getSamplingArchives(sampleName="JavaClassHistogram", outputFile="/tmp/sampling.zip")
```

The following shows the contents of the zip file:

```
unzip -l /tmp/sampling.zip
Archive:  /tmp/sampling.zip
  Length      Date    Time    Name
 --------    ----    ----    ----
  6241768  04-28-17 11:19   dfw_samplingArchive8680976839106379444.JavaClassHistogram.dmp
      552  04-28-17 11:19   dfw_samplingArchive7861027727509995202.readme.txt
 --------                   -------
  6242320                   2 files
```

For complete syntax, see getSamplingArchives in the *WLST Command Reference for Infrastructure Components*.

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 topics describe:

- Creating an Incident Manually
- Creating an Aggregated Incident
- Packaging an Incident
- Purging Incidents

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 Investigating, Reporting, and Solving a Problem.

Consider creating an incident manually when you encounter an issue, such as software failure or performance problem and you want to gather more diagnostic data, but the Diagnostic Framework has not automatically created an incident.

You use the WLST command `createIncident` to create an incident manually. You can specify an incident based on time, a message ID, an impact area, or an ECID. Then, you can inspect the content of the incident or send it to Oracle Support for further analysis.

For example, to manually create an incident based on a message ID:

1. Search the log files, as described in Searching Log Files. If you find a message that you suspect is related to the issue you are seeing, you can use the message ID when you create the incident.

2. Use the following commands to invoke WLST, connect to the Managed Server and navigate to the Managed Server instance:

```
java weblogic.WLST
connect('username', 'password', 'localhost:7001')
cd('servers/server_name')
```

3. Create the incident, using the `createIncident` command, with the following format:

```
createIncident([adrHome] [,incidentTime] [,messageId] [,ecid] [,appName]
  [,description] [,server])
```

For example, to create an incident based on the error with the message ID MDS-50500, use the following command, specifying the message ID, and provide a description of the incident to help you and Oracle support track the incident:

```
createIncident(messageId='MDS-50500', description='sample incident')
Incident Id: 1
Problem Id: 1
Problem Key: MDS-50500 [MANUAL]
Incident Time:Fri Apr 28 11:02:22 PDT 2017
Error Message Id: MDS-50500
Execution Context:null
Flood Controlled: false
Dump Files :
   jvm_threads10_i1.txt
   dms_metrics11_i1.txt
   dfw_samplingArchive13_i1.JVMThreadDump.txt
   dfw_samplingArchive13_i1.readme.txt
   odl_logs14_i1.txt
```

If you do not specify a server, the incident collects information from the server to which you are connected. To specify a server, use the `server` option, as shown in the following example:

```
createIncident(messageId='MDS-50500', description='sample incident',
server='wls_server1')
)
```

If you do not specify the `adrHome` option, the incident is created in the server to which you are connected. For example, if you are connected to the Administration Server, the incident is created in the adrHome for the Administration Server.

The Diagnostic Framework evaluates the command and invokes the appropriate diagnostic dumps. The incident and the diagnostic dumps are written to the ADR. Each diagnostic dump writes its output to the incident.

You can view the information about the incident, as described in Viewing Incidents.

You can view the information in the dumps, as described in Working with Diagnostic Dumps.

Creating an Aggregated Incident

If you have several incidents and want to combine them into a single incident, you can use the WLST `createAggregatedIncident` command. For example, if you used selective tracing, the resulting incidents containing the trace data may be generated on multiple servers. With the `createAggregatedIncident` command, you can generate an aggregated incident that meets criteria you specify. The original incidents are untouched. That is, the aggregated incident contains a copy of the incident files from the queried incidents.

The aggregated incidents are created on the Administration Server host, but they can contain incidents from one or more servers or all servers in the domain.

You construct a query using an expression that contains an incident attribute, an operator, and a string, in the following format:

```
attribute operator "string"
```

You can combine query expressions with the Boolean operators AND or OR, and group them by parentheses ().

For information about the supported attributes and operators, see createAggregatedIncident in the *WLST Command Reference for Infrastructure Components*.

Each aggregated incident will contain a zip file for each incident returned from the query, as well as descriptive text detailing the query used and the details of each incident.

For example, to create an aggregated incident for all incidents that contain the ODL_TRACE_ID of 123456 on the server wls_server1:

```
createAggregatedIncident(query="ORDL_TRACE_ID equals 123456", servers="wls_server1")
Incident 55 created, containing the following incidents:
Server wls_server1
Incident Id    Problem Key                                    Incident Time
15                TRACE [123456] [MANUAL]          Mon Apr 17 11:22:12 EDT 2017
```

To create an aggregated incident for all incidents that contain the ODL_TRACE_ID of 123456 on all servers in the domain:

```
createAggregatedIncident(query="ORDL_TRACE_ID equals 123456")
Incident 55 created, containing the following incidents:
Server wls_server1, wls_server2
Incident Id    Problem Key                                    Incident Time
15                TRACE [123456] [MANUAL]          Mon Apr 17 11:22:12 EDT 2017
```

Packaging an Incident

You can package the incident to facilitate sending the information to Oracle Support by using the ADR Command Interpreter (ADRCI). The ADRCI utility enables you to investigate and report problems in a command-line environment. With ADRCI, you can package incident and problem information into a zip file for transmission to Oracle Support.

The ADRCI command-line utility is located in the following directory:

```
(UNIX) ORACLE_HOME/oracle_common/adr
(Windows) ORACLE_HOME\oracle_common\adr
```

Packaging an incident involves a three-step process:

1. Create a logical package.

   The package is denoted as logical because it exists only as metadata in the ADR. It has no content until you generate a physical package from the logical package. The logical package is assigned a package number, and you refer to it by that number in subsequent commands.

   You can create the logical package as an empty package, or as a package based on an incident number, a problem number, a problem key, or a time interval. If you create the package as an empty package, you can add diagnostic information to it in step 2.

   Creating a package based on an incident means including diagnostic data, such as dumps, for that incident. Creating a package based on a problem number or problem key means

including in the package diagnostic data for incidents that reference that problem number or problem key. Creating a package based on a time interval means including diagnostic data on incidents that occurred in the time interval.

2. Add diagnostic information to the package.

   If you created a logical package based on an incident number, a problem number, a problem key, or a time interval, this step is optional. You can add additional incidents to the package or you can add any file within the ADR to the package. If you created an empty package, you must use ADRCI commands to add incidents or files to the package.

3. Generate the physical package.

   When you submit the command to generate the physical package, ADRCI gathers all required diagnostic files and adds them to a zip file in a designated directory. You can generate a complete zip file or an incremental zip file. An incremental file contains all the diagnostic files that were added or changed since the last zip file was created for the same logical package. You can create incremental files only after you create a complete file, and you can create as many incremental files as you want. Each zip file is assigned a sequence number so that the files can be analyzed in the correct order.

   Zip files are named according to the following format:

   ```
   packageName_mode_sequence.zip
   ```

   In the format:

   - `packageName` consists of a portion of the problem key followed by a timestamp.
   - `mode` is either `COM` or `INC`, for complete or incremental.
   - `sequence` is an integer.

For example, to package an incident, take the following steps:

1. Set the ORACLE_HOME and LD_LIBRARY_PATH environment variables as shown in the following example:

   ```
   ORACLE_HOME=ORACLE_HOME/oracle_common
   LD_LIBRARY_PATH=ORACLE_HOME/oracle_common/adr
   ```

2. Invoke ADRCI. For example:

   ```
   ORACLE_HOME/oracle_common/adr/adrci
   ```

3. Use the SET BASE command to specify the ADR Base and the SET HOMEPATH command to specify the ADR home that contains the incident. The path for the HOMEPATH is relative to the ADR Base.

   ```
   SET BASE /scratch/oracle/config/domains/wls_domain/servers/wls_server1/adr
   SET HOMEPATH diag/ofm/wls_domain/wls_server1
   ```

4. Generate the logical package:

   ```
   IPS CREATE PACKAGE INCIDENT incident_number
   ```

   For example, the following command creates a package based on incident 1:

   ```
   IPS CREATE PACKAGE INCIDENT 1
   Created package 1 based on incident id 1, correlation level typical
   ```

   ADRCI assigns the logical package a number.

5. Optionally, you can add diagnostic information to the logical package. You can add the following types of information:

- All diagnostic information for a particular incident. For example, you can add another incident that you think might be related to the incident you are packaging, using the following command:

  ```
  IPS ADD INCIDENT incident_number PACKAGE package_number
  ```

- A named file within the ADR. For example, if an incident is related to an application, you can add the .ear file for the application. You can also add a readme file with notes you provide to Oracle Support. For example, to add a file to the package, use the following command:

  ```
  IPS ADD FILE filespec PACKAGE package_number
  ```

6. Generate the physical package using the following command:

   ```
   IPS GENERATE PACKAGE package_number IN path
   ```

   For example, to generate a package with the number 1, use the following command:

   ```
   IPS GENERATE PACKAGE 1 in /tmp
   Generated package 1 in file /tmp/BEA337Web_20100223132315_COM_1.zip, mode complete
   ```

   This generates a complete physical package (zip file) in the designated path.

   For more information about ADRCI, see the ADRCI: ADR Command Interpreter chapter of *Oracle Database Utilities*.

Purging Incidents

By default, incidents are purged when the total size of all incidents exceed 500 MB. You can use the maxTotalIncidentSize MBean parameter to change this value, as described in Configuring Diagnostic Framework Settings.

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 *Oracle Database Utilities*.

Generating a RDA Report

RDA is a command-line data collector and diagnostic tool which captures the right data when requested in a single step leading to faster issue resolution. RDA can collect information such as:

- Memory, CPU and disk data
- OS patches and packages
- Environment settings
- Network configuration and statistics
- Install, version and patch information
- Product configuration and log files
- Metrics (for example Status Info, DMS Dump, MBean values)

> **Note:**
>
> RDA is quick and easy to use with the following notable characteristics:
>
> 1. Small footprint in terms of disk space, CPU, and memory usage.
> 2. Simple html menu-based interface for viewing collection included in the collection package.
> 3. Collects data passively and does not modify the state of the software or environment.
> 4. Optional oracle wallet-based credential storage (for collection logic which require a credential-based connection). For example, JDBC, WLST.
> 5. Optional security filter can be enabled to exclude potentially sensitive information.

Locating and Upgrading RDA

RDA is a common component "oracle.rda" installed in `ORACLE_HOME/oracle_common/rda`. The base "oracle.rda" version is old, upgrading "oracle.rda" to the latest version using OPatch is highly recommended.

For more information, see Resolve Problems Faster! Use Remote Diagnostic Agent (RDA) - Fusion Middleware and WebLogic Server (Doc ID 1498376.2).

Generating First Collection

If you are new to RDA, for a quick introduction to RDA's collection capabilities, try capturing an overview of the host operating system (OS):

```
# Unix
export RDA_HOME=<rda home location>
$RDA_HOME/rda.sh OS

# MS Windows
set RDA_HOME=<rda home location>
%RDA_HOME%\rda.cmd OS

# Enter an ORACLE_HOME
# If there is no Oracle product installed, hit enter

-----------------------------------------------
RDA.BEGIN: Initializes the Data Collection
-----------------------------------------------
Enter the Oracle home to be used for data analysis

# And the collection will continue and complete
```

```
-------------------------------------------
RDA Data Collection Started <timestamp>
-------------------------------------------
Processing RDA.BEGIN module ...
Processing RDA.CONFIG module ...
Processing OS.OS module ...
Processing RDA.LOAD module ...
Processing RDA.END module ...
-----------------------------------------
RDA Data Collection Ended <timestamp>
-----------------------------------------
```

An OS collection takes 30 seconds to complete. RDA captures detailed information about the operating system including data regarding memory, cpu, disk, packages, patches, and more.

Here is what RDA will generate in your working directory.

```
# Setup (result set configuration) file
output.cfg

# Result set output directory
output
   /collect
         <rda report files>

# Archive so result set can easily be shared e.g. uploaded as SR attachment
RDA_output_<hostname>.zip
```

If you wish to rerun the collection, using the newly created setup file, execute the command:

```
# Unix
$RDA_HOME/rda.sh

# MS Windows
%RDA_HOME%\rda.cmd
```

Viewing the Collection

Collection reports are written to a result set output directory. For example:

```
# Default
<working directory>/output

# Named via '-s'
<working directory>/<name specified by -s>
```

By default, RDA packages a result set output directory into an archive file as shown in the following example:

```
# Default
<working directory>/RDA_output_<host>.zip
```

```
# Named via '-s'
<working directory>/RDA_<name specified by -s>_<host>.zip
```

To view the output, open the RDA start page in a browser as follows:

```
<result set output directory>/RDA__start.htm
or
<result set output directory>/__RDAstart.htm
```

If the server where you ran RDA allows you to open files using your browser, both files can be found in the output directory. If your server does not support this, perform the steps:

1. Transfer collection archive file ("RDA_output_<host>.zip") to a machine with a browser
2. Unzip the file
3. Locate "RDA__start.htm" or "__RDAstart.htm" and double-click.

RDA start page is a basic frames-based HTML/CSS rendering of the collection. Explore the contents of the collections by clicking the links in the main and secondary index. Report content appears in the main frame on the right side.

> **Note:**
>
> The presentation layer files are all located in the result set output. There is no JavaScript, java or dependencies on remote code or artifacts.

Generating a Collection for a Fusion Middleware Product

Product specific data collection modules and profiles can be discovered as follows:

```
# All modules and profiles
# Unix
RDA_HOME/rda.sh -L

# MS Windows
RDA_HOME\rda.cmd -L

# Only Fusion Middleware modules and profiles
# Unix
RDA_HOME/rda.sh -g OFM -L

# MS Windows
RDA_HOME\rda.cmd -g OFM -L
```

If you have a WebLogic Server installed and want to capture logs, configuration, and diagnostics from a WebLogic domain, use the "OFM_Wls" profile.

```
# Unix
RDA_HOME/rda.sh -s wls_my_domain -p OFM_Wls

# MS Windows
RDA_HOME\rda.cmd -s wls_my_domain -p OFM_Wls
```

```
# Minimally setup will prompt for the following:
# - ORACLE_HOME directory
# - WebLogic Domain admin <USER> e.g. weblogic
# - Location of custom startup scripts (optional)
# RDA has logic to discover Domain Home and Servers
```

A WebLogic Server profile collection takes 5 to 10 minutes to complete. The data captured is comprehensive and includes:

- Config files, logs
  - WebLogic domain
  - Oracle installer and inventory
  - Nodemanager
- WLST reports
  - MBean metrics
  - Thread dumps
- FMW Diagnostic Framework incidents

Apart from the WebLogic Server/Domain insights, the profile also executes the OS, PERF (host performance), NET (os network), PROF (host user profile and environment) modules. The resulting collection is effectively a holistic snapshot of the targeted WebLogic implementation.

Here is what RDA will generate in your working directory.

```
# Setup (result set configuration) file
wls_my_domain.cfg

# Result set output directory
wls_my_domain
  /collect
      <rda report files>

# Archive so result set can easily be shared e.g. uploaded as SR attachment
RDA_wls_my_domain_<hostname>.zip
```

If you wish to rerun the collection, using the newly created setup file for WebLogic Server profile, execute the command:

```
# Unix
$RDA_HOME/rda.sh -s wls_my_domain

# MS Windows
%RDA_HOME%\rda.cmd -s wls_my_domain
```

Note:

The flags described in the following table are introduced in the syntax cited above:

Table 13-2 RDA Custom Setup and Profile Setting Flags

Flags	Description
`-s` <provide a result set name>	Custom name for the result set config file and output directory. "`-s`" is optional, if not used the result set will be named "output".
`-p` <provide a profile name>	A profile is a predefined modules grouping relevant to a product or problem type.

For more information to explore and understand the RDA functionality, refer to the My Oracle Support documents:

- Remote Diagnostic Agent (RDA) - Getting Started (Doc ID 314422.1)
- Resolve Problems Faster! Use Remote Diagnostic Agent (RDA) - Fusion Middleware and WebLogic Server (Doc ID 1498376.2)
- Remote Diagnostic Agent (RDA) - Information Center (Doc ID 2388643.2)

Part VI
Advanced Administration

After you perform the basic administration tasks, you may need to perform advanced administration tasks, such as managing the metadata repository and changing the network configuration of Oracle Fusion Middleware.

- Managing the Metadata Repository
- Changing Oracle Fusion Middleware Network Configurations

14

Managing the Metadata Repository

Many Oracle Fusion Middleware components use metadata repositories to hold configuration information about the component and metadata for applications.
This chapter provides information on managing the metadata repositories used by Oracle Fusion Middleware.

- About Metadata Repositories
  A metadata repository contains metadata for Oracle Fusion Middleware components, such as Oracle Application Development Framework. It can also contain metadata about the configuration of Oracle Fusion Middleware and metadata for your applications.
- Creating a Database-Based Metadata Repository
  You use the Oracle Fusion Middleware Repository Creation Utility (RCU) to create the metadata repository in an existing database.You can use RCU to create the MDS Repository or a repository for metadata for particular components. RCU creates the necessary schemas for the components.
- Managing the MDS Repository
  Oracle Metadata Services (MDS) Repository contains metadata for certain types of deployed applications. Those deployed applications can be custom Jakarta EE applications developed by your organization and some Oracle Fusion Middleware component applications, such as Oracle B2B and Oracle Web Services Manager.
- Managing Metadata Repository Schemas
  Often, you need to change the passwords of the schemas in the metadata repository to enhance security. Less often, you may need to change the character set of the repository.
- Purging Data
  When the amount of data in Oracle Fusion Middleware metadata repositories grows very large, maintaining the repositories can become difficult and can affect performance.

About Metadata Repositories

A metadata repository contains metadata for Oracle Fusion Middleware components, such as Oracle Application Development Framework. It can also contain metadata about the configuration of Oracle Fusion Middleware and metadata for your applications.

Oracle Fusion Middleware supports multiple repository types. A repository type represents a specific schema or set of schemas that belong to a specific Oracle Fusion Middleware component (for example, Oracle Application Development Framework.) Oracle Fusion Middleware supports Edition-Based Redefinition (EBR), which enables you to upgrade the database component of an application while it is in use, thereby minimizing or eliminating down time. The schemas in a repository can be EBR-enabled schemas.

A particular type of repository, the Oracle Metadata Services (MDS) Repository, contains metadata for certain types of deployed applications. This includes custom Jakarta EE applications developed by your organization and some Oracle Fusion Middleware component applications. For information related specifically to the MDS Repository type, see Managing the MDS Repository.

You can create a database-based repository or, for MDS, a database-based repository or a file-based repository. For production environments, you use a database-based repository. Some

components require that a schema be installed in a database, necessitating the use of a database-based repository. MDS supports Edition-Based Redefinition (EBR) enabled schemas.

> **Note:**
>
> After the database for the metadata repository has been used for the Oracle Fusion Middleware installation, the database service name or SID cannot be changed. However, the connect string may be changed due to High Availability or failover considerations, for example, to configure Oracle Data Guard to set up database failover.
>
> For information on setting up database failover using Oracle Data Guard for FMW infrastructure 14c RCU created metadata repository schemas, see My Oracle Support at: https://support.oracle.com/.
>
> For information on changing Oracle Fusion Middleware network configurations when moving a database to a new host, see Changing Oracle Fusion Middleware Network Configurations.

Creating a Database-Based Metadata Repository

You use the Oracle Fusion Middleware Repository Creation Utility (RCU) to create the metadata repository in an existing database.You can use RCU to create the MDS Repository or a repository for metadata for particular components. RCU creates the necessary schemas for the components.

See Repository Creation Utility Schemas, IDs, and Tablespaces in *Creating Schemas with the Repository Creation Utility* for a list of the schemas and their tablespaces and datafiles.

With RCU, you can also drop component schemas.

For information about the supported databases and the supported versions, as well as using these databases with the MDS Repository, see Supported Databases for the MDS Schema in *Oracle Fusion Middleware System Requirements and Specifications*.

> **Note:**
>
> Oracle recommends that all metadata repositories reside on a database at the same site as the components to minimize network latency issues.

For information about managing an MDS Repository, see Managing the MDS Repository.

For information about how to use RCU to create a database-based metadata repository, see About the Repository Creation Utility in *Creating Schemas with the Repository Creation Utility*.

Managing the MDS Repository

Oracle Metadata Services (MDS) Repository contains metadata for certain types of deployed applications. Those deployed applications can be custom Jakarta 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 SOA composite application (SCA).

> **See Also:**
>
> About Oracle Real Application Clusters in the *High Availability Guide* for information about using an MDS Repository with Oracle Real Application Clusters (Oracle RAC)

Overview of 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 the base can be applied without breaking customizations. When a customized application is launched, the customization content is applied over the base application.

A customizable application can have multiple customization layers. Examples of customization layers are *industry* and *site*. Each layer can have multiple customization layer values, but typically only one such layer value from each layer is applied at run time. For example, the industry layer for a customizable application can contain values for health care and financial industries; but in the deployed customized application, only one of the values from this layer is used at a time. For more information about base documents and customization layers, see

Customizing Applications with MDS in *Developing Fusion Web Applications with Oracle Application Development Framework*.

An MDS Repository can be file-based or database-based. For production environments, you use a database-based repository. You can have more than one MDS Repository for a domain.

A database-based MDS Repository provides the following features that are not supported by a file-based MDS Repository:

- Efficient query capability: A database-based MDS Repository is optimized for set-based queries. As a result, it provides better performance on such searches with the database repository.

  The MDS Repository query API provides constructs to define the query operation and to specify conditions on metadata objects. These conditions are a set of criteria that restrict the search results to a certain set of attribute types and values, component types, text content, and metadata paths. The API allows multiple conditions to be combined to achieve dynamic recursive composition using OR and AND constructs.
- Atomic transaction semantics: A database-based MDS Repository uses the database transaction semantics, which provides rollbacks of failed transactions, such as failed imports or deployments.
- Versioning: A database-based MDS Repository maintains versions of the documents in a database-based repository. Versioning allows changes to metadata objects to be stored as separate versions rather than simply overwriting the existing data in the metadata repository. It provides version history, as well as the ability to label versions so that you can access the set of metadata as it was at a given point in time.
- Isolate metadata changes: A database-based MDS Repository has the capability to isolate metadata changes in a running environment and test them for a subset of users before committing them for all users.
- Support for external change detection based on polling: This allows one application to detect changes that another application makes to shared metadata. For example, if you have an application deployed to Managed Servers A and B in a cluster, and you modify the customizations for the application deployed to Managed Server A, the data is written to the database-based repository. The application deployed to Managed Server B uses the updated customizations. This supports high availability (in particular, active/active scenarios.)
- Clustered updates: A database-based MDS Repository allows updates from multiple hosts to the metadata. For a file-based MDS Repository, updates can be made from only one host at a time.

Multiple applications can share metadata by configuring a shared metadata repository. When you do this, changes made by one application to the metadata in this repository are seen by other applications using the shared repository, if you configure external change detection for the applications.

In an MDS Repository, each application, including Oracle Fusion Middleware components, is deployed to its own partition. A **partition** is an independent logical repository within one physical MDS Repository, whether it is database-based or file-based.

For information about deploying applications and associating them with an MDS Repository, see Deploying Applications.

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.
- Databases Supported by MDS
- About MDS Operations

Databases Supported by MDS

The MDS Repository supports Oracle databases, as well as non-Oracle databases, including SQL Server, DB2, and MySQL.

For information about the supported databases and the supported versions, as well as using these databases with the MDS Repository, see Supported Databases for the MDS Schema in the *Oracle Fusion Middleware System Requirements and Specifications*

About MDS Operations

You can use Fusion Middleware Control or WLST commands to perform most operations on the MDS Repository. However, for some operations that do not have a custom user interface in Fusion Middleware Control or do not have WLST commands, you must use the System MBeans.

The sections that follow describe using Fusion Middleware Control and WLST commands to perform the operations, unless only System MBeans are supported. In that case, the sections describe how to use System MBeans to perform the operation.

You can view information about the repositories, including the partitions and the applications deployed to each partition. You can also perform operations on the partitions, such as purging, deleting, importing metadata, or exporting metadata.

Note the following when you use the MDS operations described in the sections that follow:

- The export operation exports a versioned stripe (either the tip version or based on a label) of metadata documents from an MDS Repository partition to a file system directory or archive. If you export to a directory, the directory must be accessible from the host where the application is running. If you export to an archive, the archive can be located on the system on which you are executing the command.

  Because versioning of metadata is not supported for file-based repositories, the tip version (which is also the only version) is exported from a file-based repository.
- The import operation imports metadata documents from a file system directory or archive to an MDS Repository partition. If you exported to a directory, the directory must be accessible from the host where the application is running. If you exported to an archive, the archive can be located on the system on which you are executing the command.

If the target repository is a database-based repository, the metadata documents are imported as new tip versions. If the target repository is a file-based repository, the metadata documents are overwritten.

Note:

- For more information about the custom WLST MDS commands, see Metadata Services (MDS) Custom WLST Commands in the *WLST Command Reference for Infrastructure Components*.

Table 14-1 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 14-1 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
Deregister metadata file repository	Admin role for domain
Destroy sandbox	Admin role for application
Export metadata	Monitor role for application
Export sandbox metadata	Monitor role for application
Import MAR	Admin role for application
Import metadata	Admin role for application
Import sandbox metadata	Admin role for application
List metadata label	Monitor role for application
List sandboxes	Monitor role for application
Promote metadata label	Admin role for application
Purge metadata	Admin role for application
Purge metadata labels	Admin role for application
Register metadata database repository	Admin role for domain
Register metadata file repository	Admin role for domain

For information about how these roles map to WebLogic Server roles, see Mapping of Logical Roles to WebLogic Roles in *Securing Applications with Oracle Platform Security Services*.

Registering and Deregistering a Database-Based MDS Repository

> **Note:**
>
> Note the following if you invoke the following WLST commands or comparable MBeans in a script:
>
> - registerMetadataDBRepository
> - deregisterMetadataDBRepository
>
> In this release and previous releases, the commands or MBeans have the following behavior:
>
> 1. Starts an Oracle WebLogic Server editing session.
> 2. Registers or deregisters the repository.
> 3. Activates the changes.
>
> However, you can start an editing session explicitly. If you do, the automatic activation of the changes are deprecated.

Registering a Database-Based MDS Repository

Before you can deploy an application to an MDS Repository, you must register the repository with the Oracle WebLogic Server domain. You can register a database-based MDS Repository using Fusion Middleware Control or WLST, as described in the following topics:

Registering a Database-Based MDS Repository Using Fusion Middleware Control

You create a database-based MDS Repository using RCU, as described in Creating a Database-Based Metadata Repository.

To register a database-based MDS Repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services,** then **Metadata Repositories.**

   The Metadata Repositories page is displayed.
2. In the Database-Based Repositories section, click **Register.**

   The Register Database-Based Metadata Repository page is displayed.
3. In the Database Connection section, enter the following information:
   - For **Database Type,** select the type of database.
   - For **Host Name,** enter the name of the host.

- For **Port,** enter the port number for the database, for example: `1521`.
- For **Service Name,** enter the service name for the database. The default service name for a database is the global database name, comprising the database name, such as `orcl`, and the domain name. In this case, the service name would be `orcl.domain_name.com`.
- For **User Name,** enter a user name for the database which is assigned the SYSDBA role, for example: `SYS`.
- For **Password,** enter the password for the user.
- For **Role,** select a database role, for example, **SYSDBA.**

4. Click **Query.**

   A table is displayed that shows the metadata repositories in the database.

5. Select a repository, then enter the following information at the bottom of the page:
   - For **Repository Name,** enter a name.
   - For **Schema Password**, enter the password you specified when you created the schema.
6. Click **OK.**

   The repository is registered with the Oracle WebLogic Server domain and is targeted to the Administration Server. To target the repository to other servers, see Targeting Additional Servers to an MDS Repository.

   In addition, a system data source is created with the name mds-*repository_name*. Global transaction support is disabled for the data source.

Registering a Database-Based MDS Repository Using WLST

To register a database-based MDS Repository using the command line, you use the WLST `registerMetadataDBRepository` command. You can specify the The WebLogic Server instances or clusters to which this repository will be registered. For example, to register the MDS Repository mds-repos1, to the server, server1, use the following command:

```
registerMetadataDBRepository(name='mds-repos1', dbVendor='ORACLE',
    host='hostname', port='1521', dbName='ora11',
    user='username', password='password', targetServers='server1')
```

You can specify a cluster by specifying the cluster name in the targetServers parameter.

Targeting Additional Servers to an MDS Repository

When you register an MDS Repository using Fusion Middleware Control, the repository is targeted to the Administration Server. You can target the repository to additional servers.

To target the MDS Repository to additional servers:

1. From the navigation pane, expand **Metadata Repositories.**
2. Select the repository.

   The repository home page is displayed.
3. In the Targeted Servers section, click **Add.**

   The Target the Repository dialog box is displayed.
4. Select the server or cluster and click **Target.**

You can expand the cluster to see the servers in the cluster. However, if you select a cluster, the repository is targeted to all servers in the cluster.

5. When the operation completes, click **Close.**

   The server is now listed in the Targeted Servers section.

Removing Servers Targeted to a Metadata Repository

To remove a server as a target for the repository:

1. From the navigation pane, expand **Metadata Repositories.**
2. Select the repository.

   The repository home page is displayed.
3. In the Targeted Servers section, select the target server and click **Remove.**

   The Untarget the Repository dialog box is displayed.
4. Select the server or cluster and click **Untarget.**

   You can expand the cluster to see the servers in the cluster. However, if you select a cluster, the repository will be untargeted from all servers in the cluster.
5. When the operation completes, click **Close.**

Deregistering a Database-Based MDS Repository

Deregistration does not result in loss of data stored in the repository. However, any applications using a deregistered repository will not function after the repository is deregistered. You must ensure that no application is using the repository before you deregister it.

You can deregister a database-based MDS Repository using Fusion Middleware Control or WLST, as described in the following topics:

- Deregistering a Database-Based MDS Repository Using Fusion Middleware Control
- Deregistering a Database-Based MDS Repository Using WLST

Deregistering a Database-Based MDS Repository Using Fusion Middleware Control

To deregister an MDS Repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services,** then **Metadata Repositories.**

   The Metadata Repositories page is displayed.

   Alternatively, you can navigate to the Register Metadata Repositories page by choosing **Administration,** then **Register/Deregister** from the Metadata Repository menu when you are viewing a metadata repository home page.
2. Select the repository from the table.
3. Click **Deregister.**
4. Click **Yes** in the Confirmation dialog box.

Deregistering a Database-Based MDS Repository Using WLST

To deregister a database-based MDS Repository using the command line, you use the WLST `deregisterMetadataDBRepository` command. For example, to deregister the MDS Repository mds-repos1, use the following command:

```
deregisterMetadataDBRepository(name='mds-repos1')
```

Registering and Deregistering a File-Based MDS Repository

> **Note:**
>
> Note the following if you invoke the following MBeans in a script:
>
> - registerMetadataFileRepository
> - deregisterMetadataFileRepository
>
> In this release and previous releases, the MBeans have the following behavior:
>
> 1. Start an Oracle WebLogic Server editing session.
> 2. Register or deregister the repository.
> 3. Activate the changes.
>
> However, you can start an editing session explicitly. If you do, the automatic activation of the changes are deprecated.

Creating and Registering a File-Based MDS Repository

You can create a file-based MDS Repository and register it with an Oracle WebLogic Server domain using Fusion Middleware Control.

To create and register a file-based repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services,** then **Metadata Repositories.**

   The Metadata Repositories page is displayed.
2. In the File-Based Repository section, click **Register.**

   The Register Metadata Repository page is displayed.
3. Enter the following information:
   - For **Name,** enter a name. For example, enter repos1. The prefix `mds-` is added to the name and a repository with the name mds-repos1 is registered. If you enter a name that begins with `mds-`, a repository with the given name is registered.
   - For **Directory,** specify the directory. The Administration Server and Managed Servers that run the applications that use this repository must have write access to the directory.

     Note the following:

     – If an absolute path is not given, the directory will be created under the *DOMAIN_HOME* directory.

     – If the specified path exists on the file system, the metadata file repository is registered; all the subdirectories under this path are automatically loaded as partitions of this file-based repository.

- If the path specified does not exist, a directory with this name is created on the file system during the registration. Because there are no partitions created yet, there are no subdirectories to load.
- If the specified path is invalid and cannot be created for some reason, such as permission denied, an error is displayed and the registration fails.
- If the specified path exists, but as a file not a directory, an error is not displayed and the registration succeeds.

4. From **Scope,** select Global or a name of a partition.
5. Click **OK.**

The repository is created and registered and is displayed on the Metadata Repositories page.

You can now create and delete partitions. Those changes are reflected in the directory on the file system.

You can also create a file-based repository using system MBeans. For information about using the System MBean Browser, see Using System MBeans to Manage an MDS Repository.

Deregistering a File-Based MDS Repository

You can deregister a file-based MDS Repository using Fusion Middleware Control.

To deregister a file-based repository using Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Other Services,** then **Metadata Repositories.**

   The Metadata Repositories page is displayed.
2. In the File-Based Repository section, select the repository and click **Deregister.**
3. Click **OK** in the Confirmation dialog box.

   If the file-based repository is valid, it is removed from the repository list. Otherwise, an error is displayed.

You can also deregister a file-based repository using system MBeans. For information about using the System MBean Browser, see Using System MBeans to Manage an MDS Repository.

Changing the System Data Source

You can change the system data source to reassociate an application to a new repository. You can change the database or the schema that contains the data source. To do so, you can use Oracle WebLogic Remote Console or Fusion Middleware Control. To use Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **JDBC Data Sources.**

   The JDBC Data Sources page is displayed.
2. Select the data source you want to change.

   The JDBC Data Source page for the selected data source is displayed.
3. Select the Connection Pool tab.
4. To change the database, modify the **Database URL** field. For example:

   `jdbc:oracle:thin:@hostname.domainname.com:1522/orcl`
5. For **Password,** enter the password for the database.
6. For **Confirm Password,** reenter the password for the database.

7. To change the schema, modify the Properties section, changing the value for **user.**
8. If the database is a DB2 database, add the property sendStreamAsBlob, with a value of true.
9. Click **Save.**
10. Restart the servers that use this data source.

Using System MBeans to Manage an MDS Repository

Although most procedures in this chapter discuss using Fusion Middleware Control or WLST to manage the MDS Repository, you can also use system MBeans:

1. From the WebLogic Domain menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.
2. In the page's navigation pane, expand **Application Defined MBeans,** then expand **oracle.mds.lcm.** Expand the domain, then **MDSDomainRuntime,** and then select **MDSDomainRuntime.**
3. In the Application Defined MBeans pane, select the Operations tab.
4. Click one of the operations, such as **registerMetadataFileRepository.**

   The Operations page is displayed.
5. In the Value column, enter values for the operation.
6. Click **Invoke.**

Viewing Information About an MDS Repository

You can view information about an MDS Repository using Fusion Middleware Control or system MBeans, as described in the following topics:

- Viewing Information About an MDS Repository Using Fusion Middleware Control
- Viewing Information About an MDS Repository Using System MBeans

Viewing Information About an MDS Repository Using Fusion Middleware Control

To view information about an MDS Repository using Fusion Middleware Control:

1. From the navigation pane, expand **Metadata Repositories.**
2. Select the repository.
3. To see which applications use the repository, click the icon in the Applications column. The Applications using the partition dialog box is displayed:
   - The Deployed Applications tab shows the list of applications whose metadata is deployed to the repository partition.
   - The Referenced by Applications tab shows the list of applications that refer to the metadata stored in the repository partition.

From this page, you can also:

- Delete partitions, as described in Deleting a Metadata Partition Using Fusion Middleware Control.
- Manage labels, as described in Deleting Metadata Labels.

- Add or remove targeted servers, as described in Targeting Additional Servers to an MDS Repository.

Viewing Information About an MDS Repository Using System MBeans

You can use the System MBean operations listPartitions, listRepositories, and listRepositoryDetails to get a list of partitions in the repository, a list of repositories, and details of the repository registered with the domain:

1. From the WebLogic Domain menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.

2. In the page's navigation pane, expand **Application Defined MBeans,** then expand **oracle.mds.lcm.** Expand the domain, then **MDSDomainRuntime,** and then select **MDSDomainRuntime.**

3. In the Application Defined MBeans pane, select the Operations tab.

4. Click one of the operations, such as listPartitions, listRepositories, and listRepositoryDetails.

   The Operations page is displayed.

5. Click **Invoke.**

   The information is displayed in the Return Value table.

For information about changing the MDS configuration attributes for an application, see Changing MDS Configuration Attributes for Deployed Applications.

Configuring an Application to Use a Different MDS Repository or Partition

When you deploy an application, you can associate it with an MDS Repository. You can subsequently change the MDS Repository or partition to which an application is associated, using WLST or Fusion Middleware Control. For example, a different repository contains different metadata that needs to be used for a particular application.

To associate an application with a new MDS Repository or partition, you can either:

- Redeploy the application, specifying the new repository or partition.

  To create a new partition, you can either:

  – Clone the partition to a different repository. Cloning the partition is valid only with a database-based repository with databases of the same type and version. When you clone the partition, you preserve the metadata version history, including any customizations and labels.

    Cloning a Partition describes how to clone a partition and how to redeploy the application, specifying the partition that you have cloned.

  – Create a new partition, then export the metadata from the current partition and import the metadata into the new partition.

    Creating a New Partition and Reassociating the Application to It describes how to create the partition and export and import data and how to redeploy the application, specifying the new repository or partition.

- Change the system data source. When you change the system data source, you can change the database or the schema in which it is stored.

  Changing the System Data Source describes how to change the system data source.

- Cloning a Partition
- Creating a New Partition and Reassociating the Application to It

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.
   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 Redeploying Oracle ADF Applications or Redeploying SOA Composite Applications, 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.**

Creating a New Partition and Reassociating the Application to It

You can create a new partition in the same or a different repository by redeploying the application and specifying the new partition. Then, you transfer the metadata to the new partition using WLST.

You can use this procedure to transfer metadata between two different types of repositories (file-based to database-based or from an Oracle Database to another database.)

To create a new partition and reassociate the application to it:

1. Export the metadata from the source partition to a directory on the file system using the WLST `exportMetadata` command:

   ```
   exportMetadata(application='sampleApp', server='server1',
          toLocation='/tmp/myrepos/mypartition', docs='/**')
   ```

2. Redeploy the application, as described in Redeploying Oracle ADF Applications or Redeploying SOA Composite Applications, 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 Deregistering a File-Based MDS Repository or Deregistering a Database-Based MDS Repository.

Alternatively, you can create a new partition using the WLST command `createMetadataPartition`. The partition name must be unique within the repository. If the partition parameter is missing, the name of the source partition is used for the target partition. The following example creates the partition partition1:

```
createMetadataPartition(repository='mds-repos1', partition='partition1')
```

Moving Metadata from a Source System to a Target System

You can transfer the metadata in MDS from one partition to another. As an example, you want to move an application from a test system to a production system. You have a test application that is deployed in a domain in the test system and a production application deployed in a domain in the production system. You want to transfer the customizations from the test system to the production system. To do that, you transfer the metadata from the partition in the test system to a partition in the production system.

To transfer the metadata from one partition to another, you export the metadata from the partition and then import it into the other partition. You can use Fusion Middleware Control or WLST to transfer the metadata, as described in the following topics:

- Transferring Metadata Using Fusion Middleware Control
- Transferring Metadata using WLST

Transferring Metadata Using Fusion Middleware Control

To use Fusion Middleware Control to transfer metadata:

1. From the WebLogic Domain menu, select **Deployments.**
2. Select the application.
3. From the Application Deployment menu, choose **MDS Configuration.**

   The MDS Configuration page is displayed.
4. In the Export section, select one of the following:
   - **Export metadata documents to an archive on the machine where this web browser is running.**

     Click **Export.**

     The export operation exports a zip file. Depending on the operating system and browser, a dialog box is displayed that asks you if you want to save or open the file.
   - **Export metadata documents to a directory or archive on the machine where this application is running.**

     Enter a directory location or archive to which the metadata can be exported.

     The target directory or archive file (.jar, .JAR, .zip or .ZIP) to which to transfer the documents selected from the source partition. If you export to a directory, the directory must be a local or network directory or file where the application is physically deployed. If you export to an archive, the archive can be located on a local or network directory or file where the application is physically deployed, or on the system on which you are executing the command.

     If the location does not exist in the file system, a directory is created except that when the names ends with .jar, .JAR, .zip or .ZIP, an archive file is created. If the archive file already exists, the exportMetadata operation overwrites the file.

     Click **Export.** Then, in the Confirmation dialog box, click **Close.**

   If you check **Exclude base documents,** this operation exports only the customizations, not the base documents. See Overview of the MDS Repository for information about base documents and customizations.
5. If the target application is on a different system, copy the exported metadata to that system.
6. On the target system, from the WebLogic Domain menu, select **Deployments.**
7. Select the application.
8. From the Application Deployment menu, choose **MDS Configuration.**

   The MDS Configuration page is displayed
9. In the Import section, select one of the following:
   - **Import metadata documents from an archive on the machine where this web browser is running.**

     Click **Browse** and select the file.
   - **Import metadata documents from a directory or archive on the machine where this application is running.**

     Enter the location of the directory or archive that contains the exported metadata. If you specify a directory, include the subdirectory with the partition name in the

specification. The directory or archive file must be a local or network directory or file where the application is physically deployed.

10. Click **Import.**
11. In the Confirmation dialog box, click **Close.**

Transferring Metadata using WLST

To use WLST to transfer metadata:

1. Export the metadata from the original partition using the `exportMetadata` command:

```
exportMetadata(application='sampleApp', server='server1',
       toLocation='/tmp/myrepos/mypartition', docs='/**')
```

   This command exports a versioned stripe of the metadata documents from the metadata partition to a file system directory. Only customization classes declared in the cust-config element of adf-config.xml are exported. If there is no cust-config element declared in adf-config.xml, all customization classes are exported.

   To export all customizations, use the option restrictCustTo="%".

2. If the production application is on a different system, copy the exported metadata to that system.
3. Import the metadata to the other partition using the WLST `importMetadata` command:

```
importMetadata(application='sampleApp', server='server1',
       fromLocation='/tmp/myrepos/mypartiton', docs='/**')
```

   The value of the `fromLocation` parameter must be on the same system that is running WLST or on a mapped network drive or directory mount. You cannot use direct network references such as `\\mymachine\repositories\`.

   Only customization classes declared in the cust-config element of adf-config.xml are imported. If there is no cust-config element declared in adf-config.xml, all customization classes are imported.

   To import all customizations, use the option restrictCustTo="%".

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 Creating a Database-Based Metadata Repository.
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 Redeploying Oracle ADF Applications or Redeploying SOA Composite Applications, 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 Deregistering a File-Based MDS Repository.

Deleting a Metadata Partition from a Repository

You can delete metadata partitions if there are no applications either deployed to the partition or referring to the partition. You may want to delete a metadata partition from the repository in the following circumstances:

- When you undeploy an application. Oracle Fusion Middleware leaves the metadata partition because you may still want the metadata, such as user customizations, in the partition. If you do not need the metadata, you can delete the partition.
- When you have transferred metadata from one partition to another and configured the application to use the new partition.
- When you have cloned a partition and configured the application to use the new partition.

Note that deleting a partition deletes all the data contained in the partition.

You can delete a metadata partition using WLST or Fusion Middleware Control, as described in the following topics:

- Deleting a Metadata Partition Using Fusion Middleware Control
- Deleting a Metadata Partition Using WLST

Deleting a Metadata Partition Using Fusion Middleware Control

To delete a metadata partition from a repository partition using Fusion Middleware Control:

1. From the navigation pane, expand **Metadata Repositories.**
2. Select the repository.

   The repository home page is displayed.
3. In the Repository Partitions section, select the partition and click **Delete.**
4. In the confirmation dialog box, click **OK.**

Deleting a Metadata Partition Using WLST

To delete a metadata partition from a repository, you can use the WLST command `deleteMetadataPartition`. For example, to delete the metadata partition from the file-based repository mds-repos1, use the following command:

```
deleteMetadataPartition(repository='mds-repos1', partition='partition1')
```

Purging Metadata Version History

For database-based MDS Repositories, you can purge the metadata version history from a partition. (File-based MDS Repositories do not maintain version history.) This operation purges version history of unlabeled documents from the application's repository partition. The tip version (the latest version) is not purged, even if it is unlabeled.

To purge metadata labels, you use the `purgeMetadataLabels` command, as described in Purging Metadata Labels. Then, you can purge the metadata version history.

Consider purging metadata version history on a regular basis as part of MDS Repository maintenance, when you suspect that the database is running out of space or performance is becoming slower. This operation may be performance intensive, so plan to do it in a maintenance window or when the system is not busy. Note that MDS purges 300 documents in each iteration, commits the change, and repeats until all purgeable documents are processed.

For specific recommendations for particular types of applications, see the documentation for a particular component.

You can purge metadata version history using WLST or Fusion Middleware Control, as described in the following topics:

- Purging Metadata Version History Using Fusion Middleware Control
- Purging Metadata Version History Using WLST
- Enabling Auto-Purge

Purging Metadata Version History Using Fusion Middleware Control

To use Fusion Middleware Control to purge the metadata version history:

1. From the navigation pane, expand **Application Deployments,** then select the application.
2. From the Application Deployment menu, choose **MDS Configuration.**

   For Oracle SOA Suite, you can expand **SOA** in the navigation tree, then select **soa-infra.** From the SOA Infrastructure menu, select **Administration,** then **MDS Configuration.**

   The MDS Configuration page is displayed.
3. In the Purge section, in the **Purge all unlabeled past versions older than** field, enter a number and select the unit of time. For example, enter **3** and select **months.**
4. Click **Purge.**
5. In the Confirmation dialog box, click **Close.**

Purging Metadata Version History Using WLST

To use WLST to purge metadata version history, use the `purgeMetadata` command. You specify the documents to be purged by using the `olderThan` parameter, specifying the number of seconds. The following example purges all documents older than 100 seconds:

```
purgeMetadata(application='sampleApp', server='server1', olderThan=100)
```

Enabling Auto-Purge

You can enable automatic purging using the MDSAppConfig MBean:

1. From the WebLogic Domain menu, choose **System MBean Browser.**

   The System MBean Browser page is displayed.
2. Expand **Application Defined MBeans,** then **oracle.adf.share.config,** then **Server: *name***, then **Application: *name***, then **ADFConfig,** then **ADFConfig,** and **ADFConfig.**
3. Select **MDSAppConfig.**

   The Application Defined MBeans page is displayed.
4. For **AutoPurgeTimeToLive,** enter a value, in seconds.
5. Navigate up to ADFConfig (the parent of MDSAppConfig) and select it.
6. In the Operations tab, click **Save.**

Managing Metadata Labels in the MDS Repository

About Metadata Labels

A **metadata label** is a means of selecting a particular version of each object from a metadata repository partition. Conceptually, it is a collection of document versions, one version per document, representing a *horizontal stripe* through the various document versions. This stripe comprises the document versions which were the tip versions (latest versions) at the time the label was created.

You can use a label to view the metadata as it was at the point in time when the label was created. You can use the WLST commands to support logical backup and recovery of an application's metadata contained in the partition.

Labels are supported only in database-based repositories.

Document versions belonging to a label are not deleted by automatic purging, unless the label is explicitly deleted. In this way, creating a label guarantees that a view of the metadata as it was at the time the label was created remains available until the label is deleted.

When an application that contains a MAR is deployed, a label with the prefix `postDeployLabel_` is created. For example: postDeployLabel_mdsappdb_mdsappdb.mar_2556916398.

Each time you patch the MAR, a new deployment label is created, but the previous deployment label is not deleted Similarly, when you undeploy an application that contains a MAR, the application is undeployed, but the label remains in the metadata repository partition.

If you delete a deployment label, when the application is restarted, the MAR is automatically redeployed, and the deployment label is also re-created.

Creating Metadata Labels

To create a label for a particular version of objects in a partition in an MDS Repository, you use the WLST command `createMetadataLabel`. For example, to create a label named prod1 for the application my_mds_app, use the following command:

```
createMetadataLabel(application='my_mds_app', server='server1', name='prod1')
Executing operation: createMetadataLabel.

Created metadata label "prod1".
```

If the application has more than one version, you must use the applicationVersion parameter to specify the version.

Listing Metadata Labels

You can list the metadata labels for a particular application. To do so, use the WLST command `listMetadataLabel`. For example, to list the labels for the application my_mds_app, use the following command:

```
listMetadataLabels(application='my_mds_app', server='server1')
Executing operation: listMetadataLabels.

Database Repository partition contains the following labels:
prod1
prod2
postDeployLabel_mdsappdb_mdsappdb.mar_2556916398
```

If the application has more than one version, you must use the applicationVersion parameter to specify the version.

Promoting Metadata Labels

You can promote documents associated with a metadata label so that they become the latest version. That is, you can promote them to the tip. Promote a label if you want to roll back to an earlier version of all of the documents captured by the label.

To promote a label to the tip, use the WLST command `promoteMetadataLabel`. For example to promote the label prod1, use the following command:

```
promoteMetadataLabel(application='my_mds_app', server='server1', name='prod1')
Location changed to domainRuntime tree. This is a read-only tree with DomainMBean as the
root.
For more help, use help(domainRuntime)

Executing operation: promoteMetadataLabel.

Promoted metadata label "prod1" to tip.
```

If the application has more than one version, you must use the applicationVersion parameter to specify the version.

Purging Metadata Labels

You can purge metadata labels that match the given name pattern or age, allowing you to purge labels that are no longer in use. This reduces the size of the database, improving

performance. You must delete the labels associated with unused metadata documents before you can purge the documents and revision history from the repository.

You may want to delete a label for older applications that were undeployed, but the labels were not deleted. Each time you patch the MAR, a new label is created, but the previous label is not deleted.

You can use Fusion Middleware Control or WLST to purge metadata labels, as described in the following topics:

- Purging Metadata Labels Using Fusion Middleware Control
- Purging Metadata Labels Using WLST

Purging Metadata Labels Using Fusion Middleware Control

To purge metadata labels using Fusion Middleware Control:

1. From the navigation pane, expand **Metadata Repositories.**
2. Select the repository.

   The repository home page is displayed.
3. Select a partition and click **Manage Labels.**

   The Manage Labels page is displayed. By default, the table lists all metadata labels created in the selected partition that are more than one year old and that are not deployed or associated with a sandbox.
4. To search for a particular label or labels, you can:
   - For **Label Name,** select an operator and enter the filter criteria. The characters are case sensitive. You can use the following wildcards:
     - Percent (%): Matches any number of characters
     - Underscore (_): Matches a single character
     - Backslash (\): Used as an escape character for the wildcards

     For example, the string `postDeployLabel%` returns any label beginning with `postDeployLabel`. As a result, it displays labels associated with a deployed MAR.
   - For **Age,** enter a number, such as 2. (The only operator available is Older Than.)
   - For **Age (units),** select a unit, such as Hours, Days, Weeks, Months, Years. The only operator available is Equals.
5. Click **Search.**
6. By default, labels associated with sandboxes and deployed applications are not shown. To display those labels, select **Sandboxes** or **Deployment** or both. Note the following:
   - You cannot delete a label associated with a sandbox.
   - If you select **Deployment,** the labels that are associated with MAR deployments are displayed.
7. Select the label and click **Delete Selected.**
8. In the confirmation box, click **OK.**

If you want to purge all unused labels, for a particular deployed application:

1. Select **Deployment.**
2. Filter by name, using the string `postDeployLabel_`*`application_name`*`%`.

3. Select all but the latest (which is in use) to delete. (The most recent label---the one that is currently being used---is listed first.)
4. Click **Delete Selected.**

Purging Metadata Labels Using WLST

You can purge metadata labels that match the given pattern or age, using the WLST command `purgeMetadataLabels`. The command purges the labels that match the criteria specified, but it does not delete the metadata documents that were specified by the labels.

For example, to purge all metadata labels that match the specified namePattern and that are older than 30 minutes:

```
purgeMetadataLabels(repository='mds-myRepos', partition='partition1',
                    namePattern='prod*', olderThanInMin='30')
Location changed to domainRuntime tree. This is a read-only tree with DomainMBean as the
root.
For more help, use help(domainRuntime)

Executing operation: purgeMetadataLabels.

The following metadata labels were purged: repository=mds-
soa,parititon=partition1,namePattern=prod*,olderThanInMin=30:
```

Deleting Metadata Labels

To delete a specified metadata label, you use the WLST command `deleteMetadataLabel`. For example, to delete a label named prod1 for the application my_mds_app, use the following command:

```
deleteMetadataLabel(application='my_mds_app', server='server1', name='prod1')
```

If the application has more than one version, you must use the applicationVersion parameter to specify the version.

To find the labels associated with an application, use the `listMetadataLabels` command, as described in Listing Metadata Labels.

Managing Metadata Repository Schemas

Often, you need to change the passwords of the schemas in the metadata repository to enhance security. Less often, you may need to change the character set of the repository.

Changing Metadata Repository Schema Passwords

The schema passwords are stored in the database. Note that passwords expire after a period of time. For example, for an 11*g* Oracle Database, by default, the passwords expire after 180 days.

For most components, you only need to change the password in the database. However, for Oracle Platform Security Services, you need to take additional steps.

- Changing the Schema Password for Oracle Platform Security Services

Changing the Schema Passwords for Most Components

To change the schema password of most components, you change the password 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;
   COMMIT;
   ```

   For example, to change the OFM_ MDS password to `abc123`:

   ```
   SQL> ALTER USER OFM_MDS IDENTIFIED BY abc123;
   COMMIT;
   ```

3. If you change the MDS Repository schema password, you must change the password for the corresponding MDS Repository data source, using Fusion Middleware Control:
   a. From the WebLogic Domain menu, select **JDBC 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.

Changing the Schema Password for Oracle Platform Security Services

To change the schema password for Oracle Platform Security Services:

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;
   COMMIT;
   ```

   Be sure to issue the commit command before proceeding to the next step.
3. Run the WLST command modifyBootStrapCredential to update the JPS configuration file.
   a. Invoke WLST from the following directory:

      ```
      ORACLE_HOME/oracle_common/common/bin/wlst.sh
      ```

   b. Specify the full path to the JPS configuration file in the modifyBootStrapCredentials command. For example:

      ```
      modifyBootStrapCredential(jpsConfigFile='/scratch/oracle//config/domains/
      soa_domain/config/fmwconfig/jps-
      config.xml',username='schema_username',password='password')
      ```

      At this point, the Administration Server can be started, however, the log file will show the following exception:

```
####<Jun 01, 2017 2:15:09 PM CEST> <Error> <Deployer> <deployer> <AdminServer>
<[ACTIVE] ExecuteThread: '3' for queue: 'weblogic.kernel.Default
(self-tuning)'> <<WLS Kernel>> <>
<f9d07f66-36d0-462e-83fd-6ca40ac15a8a-00000004> <1402936508655> <BEA-149205>
<Failed to initialize the application "opss-data-source" due to error
weblogic.application.ModuleException:
weblogic.common.resourcepool.ResourceSystemException:
Could not connect to 'oracle.jdbc.OracleDriver'.

The returned message is: ORA-01017: invalid username/password; logon denied.
```

To avoid this error, execute next step.

4. Update the data source configuration, as described in Changing the Schema Passwords for Most Components, step 3.

Changing the Character Set of the Metadata Repository

For information about changing the character set of metadata repository that is stored in an Oracle Database, see *Oracle Database Globalization Support Guide*:

http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html

Oracle recommends using Unicode for all new system deployments. Deploying your systems in Unicode offers many advantages in usability, compatibility, and extensibility. Oracle Database enables you to deploy high-performing systems faster and more easily while utilizing the advantages of Unicode. Even if you do not need to support multilingual data today, nor have any requirement for Unicode, it is still likely to be the best choice for a new system in the long run and ultimately saves time and money and gives you competitive advantages in the long term.

When storing the metadata in a SQL Server database, if the character set being considered for your locale is not case neutral, the case-sensitive collation must be selected during the creation of the database instance. Unicode support is the default when creating the MDS schema for SQL Server using RCU. You may overwrite this default to use non-unicode schema if that meets your requirements.

Purging Data

When the amount of data in Oracle Fusion Middleware metadata repositories grows very large, maintaining the repositories can become difficult and can affect performance.

Purging Data Documentation

In some cases, Oracle Fusion Middleware automatically purges data from the repositories. In other cases, Oracle Fusion Middleware provides methods to manage growth, including scripts to purge data that can accumulate over time and that can affect performance.

Many of the Oracle Fusion Middleware components provide scripts written as PL/SQL procedures to purge the data. The scripts are located in:

ORACLE_HOME/common/sql/*component-name*_purge_*purgetype*.sql

For example, a script that purges logs for Oracle Business Process Management is located in:

ORACLE_HOME/common/sql/bpm_purge_logs.sql

#unique_402/unique_402_Connect_42_CIHIJBHB provides pointers to information about purging data for Oracle Fusion Middleware components.

Component	Description
MDS Repository	See Purging Metadata Version History for information on automatically and manually purging data.
Oracle Application Development Framework	See Cleaning Up Temporary Storage Tables in *Developing Fusion Web Applications with Oracle Application Development Framework*.
Oracle Application Development Framework Business Components	Use the following script to purge rows in the database used by Oracle ADF Business Components to store user session state and temporary persistent collections: `ORACLE_HOME/oracle_common/common/sql/adfbc_purge_statesnapshots.sql` The PS_TXN table is automatically purged.
Oracle BI Enterprise Edition	No configuration required. Automatically purges data.
Oracle Business Intelligence Publisher	Delete job history, as described in Deleting a Job History in the *User's Guide for Oracle Business Intelligence Publisher*.
Oracle Web Services Manager	No configuration required. Automatically purges data.
Oracle WebCenter Content	Export the data with deletion, as described in Exporting Data in Archives. Then, remove the collection, as described in Removing a Collection. Both sections are in the *Administering Oracle WebCenter Content*.
Oracle WebCenter Portal Analytics	See Partitioning Oracle WebCenter Portal's Analytics Data .
Oracle WebCenter Portal's Activity Stream	See Purging Oracle WebCenter Portal's Activity Stream Data.
Oracle WebLogic Server: JAXWS Web Services	Clean up the Web service persistence store, as described in Cleaning Up Web Service Persistence in *Developing JAX-WS Web Services for Oracle WebLogic Server*. Use the defaultMaximumObjectLifetime field of the WebServicePersistenceMBean to set the maximum lifetime of the objects. See Understanding WebLogic Server MBeans in *Developing Custom Management Utilities Using JMX for Oracle WebLogic Server*.
Oracle WebLogic Server: JMS	See Configuring Basic JMS System Resources and Managing JMS Messages in *Administering JMS Resources for Oracle WebLogic Server*. Also see Tuning WebLogic JMS in *Tuning Performance of Oracle WebLogic Server*.
Oracle WebLogic Server: Oracle Infrastructure Web Services	Use the following script to purge data if WS-RM uses a database store: `ORACLE_HOME/oracle_common/common/sql/ows_purge_wsrm_msgs.sql`
Oracle WebLogic Server: Session persistence for JDBC or file-based data sources	No configuration required. Automatically purges data.
Oracle WebLogic Server: Stateful EJBs	No configuration required. Automatically purges data.

In certain circumstances, you can consider using Oracle Scheduler to automate the running of the scripts. For example, you may want to set up a scheduled job to purge the last 14 days for completed instances.

In certain circumstances, you can consider using Oracle Scheduler to automate the running of the scripts. For example, you may want to set up a scheduled job to purge the last 14 days for completed instances for Oracle SOA Suite.

Oracle Scheduler, an enterprise job scheduler, is part of Oracle Database. Oracle Scheduler is implemented by the procedures and functions in the DBMS_SCHEDULER PL/SQL package. For information about Oracle Scheduler, see Oracle Scheduler Concepts and Creating, Running, and Managing Jobs in the *Oracle Database Administrator's Guide*.

Purging Oracle Infrastructure Web Services Data

Use the following script to purge data if WS-RM uses a database store:

ORACLE_HOME/oracle_common/common/sql/ows_purge_wsrm_msgs.sql

Purging Oracle WebCenter Portal Data

Purging Oracle WebCenter Portal's Activity Stream Data

Oracle WebCenter Portal's Activity Stream provides a set of WLST commands for purging database records in a nonpartitioned environment. Purging is necessary when a database contains records that are not needed as an analysis in reports or when the performance of Oracle WebCenter Portal decreases because of the large volume of data.

To purge Oracle WebCenter Portal's Activity Stream data, you use the following WLST commands:

- archiveASByDate: Archives activity stream data that is older than a specified date.
- archiveASByDeletedObjects: Archives activity stream data associated with deleted objects
- archiveASByClosedSpaces: Archives activity stream data associated with Spaces that are currently closed.
- archiveASByInactiveSpaces: Archives activity stream data associated with Spaces that have been inactive since a specified date.
- restoreASByDate: Restores archived activity stream data from a specified date into production tables.

For more information about these commands, see Activity Stream in the *WebCenter WLST Command Reference*.

Purging Oracle WebCenter Portal's Analytics Data

Oracle WebCenter Portal's Analytics provides a script for purging database records in a nonpartitioned environment. Purging is necessary when a database contains records that are

not needed for analysis in reports or when the performance of Oracle WebCenter Portal decreases because of the large volume of data.

The script, analytics_purge_facts.sql, deletes all fact tables that meet the specified criteria.

When Oracle WebCenter Portal's Analytics runs in a partitioned environment, you should use the drop partitioning feature of the database before running these scripts.

- Loading the Oracle WebCenter Portal Purge Package
- Running the Oracle WebCenter Portal Purge Script

Loading the Oracle WebCenter Portal Purge Package

Before you run the script for the first time, you must install the purge package into the database by running the analytics_purge_package script:

1. Log in to the database as the schema user for the ACTIVITIES schema.
2. Execute the analytics_purge_package script. For example, for an Oracle Database:

   ```
   @ORACLE_HOME/oracle_common/common/sql/oracle/analytics_purge_package.sql
   ```

   For a DB2 database, use the following command:

   ```
   db2 -td@ -f analytics_purge_package.sql
   ```

Running the Oracle WebCenter Portal Purge Script

The location of the analytics_purge_facts.sql script differs depending on the type of database used:

- Oracle Database:

  ```
  ORACLE_HOME/oracle_common/common/sql/oracle/analytics_purge_facts.sql
  ```

- SQL Server:

  ```
  ORACLE_HOME/oracle_common/common/sql/sqlserver/analytics_purge_facts.sql
  ```

- DB2:

  ```
  ORACLE_HOME/oracle_common/common/sql/db2/analytics_purge_facts.sql
  ```

The analytics_purge_facts.sql script takes the following parameters:

- Month From: The script purges data that was created after the beginning of the specified month. Enter the month in the format MM. For example, 08 to specify August.
- Year From: With the Month From parameter, the script purges data that was created after the beginning of the specified month in the specified year. Enter the year in YYYY format. For example, 2017.
- Month To: The script purges data that was created through the end of the specified month. Enter the month in the format MM. For example, if you specify 09 for September, the script purges all data that was created before the end of September.
- Year To: With the Month To parameter, the script purges data that was created through the end of the specified month in the specified year. Enter the year in YYYY format. For example, 2017.
- Record Batch Size: The maximum size of records to commit at one time.
- Max Run Time: The maximum amount of time, in minutes, that the you want the process to run. When the process reaches this time, it stops, regardless of the progress of the purge.

> **Note:**
>
> You cannot delete the current month. If you specify the current month, the script returns an error.

When you are using an Oracle Database or a DB2 database, the script prompts you for input for each parameter.

When you are using a SQL Server database, you must edit the analytics_purge_facts.sql script to specify the criteria for purging data.

The following shows an example of the script for SQL Server that deletes all Analytics fact database records from February 1, 2017 through May 31, 2017:

```
CALL ANALYTICS_PURGE
        (
        2, --from month
        2017, --from year
        5, --to month
        2017, --to_year
        1000, --commit batch size
        60 --max run time minutes
        );
```

To use the script:

1. If you are using a SQL Server database, edit the script to specify the criteria.
2. Execute the script. For example, to execute the script on an Oracle Database:

```
sqlplus analytics_user/analytics_user_pwd @analytics_purge_facts.sql
Enter value for month_from: 2
old   4: ANALYTICS_PURGE.PURGE_ANALYTICS_INSTANCES ( &month_from,
-- MM format
new   4: ANALYTICS_PURGE.PURGE_ANALYTICS_INSTANCES ( 2,     -- MM format
Enter value for year_from: 2017
old   5:                           &year_from,              -- YYYY format
new   5:                           2017,                    -- YYYY format
Enter value for month_to: 5
old   6:                           &month_to,               -- MM format
new   6:                           5,                      -- MM format
Enter value for year_to: 2017
old   7:                           &year_to,                -- YYYY format
new   7:                           2017,                    -- YYYY format
Enter value for record_commit_batch_size: 1000
old   8:                           &record_commit_batch_size,
new   8:                           1000,
Enter value for max_minutes_run: 60
old  10:                           &max_minutes_run) ;
new  10:                             60) ;
Log (06-01-2017 08:27:49) Purge Process Started
 .
 .
 .
Log (06-01-2017 08:27:49)
Log (06-01-2017 08:27:49) Purge Process Finished

PL/SQL procedure successfully completed.
```

Partitioning Oracle WebCenter Portal's Analytics Data

When you use the Oracle Fusion Middleware Repository Creation Utility (RCU) to create schemas, you can specify that Activity Graph and Analytics tables are partitioned (see the Custom Variables screen in RCU). If you chose to partition the tables, Oracle WebCenter Portal uses the native partitioning of the database to automatically create partitions.

Oracle WebCenter Portal provides a partition manager process, which runs once every 24 hours as a separate thread. It creates partitions on each Analytics fact table (ASFACT_*) in the database. Initially, the process generates six partitions in advance, with each partition corresponding to a month in the future. Whenever a new month starts, the partition manager creates a new partition.

Partitioning the data makes it easier to purge data, because you can purge the data by dropping the older partitions that the partition manager creates. Thus, in a partitioned environment, the recommended method for purging data is simply to drop the month-based partitions that are no longer required.

> **Note:**
>
> The WC_Utilities Managed Server must be started for the partition manager process to run.

For example, to drop older partitions for a table, use the following SQL command:

```
alter table table_name drop partition partition_name;
```

15

Changing Oracle Fusion Middleware Network Configurations

Oracle Fusion Middleware provides procedures for changing the network configuration, such as the host name or IP address, of an Oracle Fusion Middleware host and the Oracle database that Oracle Fusion Middleware uses. It also provides support for the IPv6 protocol with Oracle Fusion Middleware.

- About Changing the Network Configuration
  You can change the host name and IP address of an Oracle Fusion Middleware domain and the database that is used by Oracle Fusion Middleware.
- Moving Between On-Network and Off-Network
  You can move an Oracle Fusion Middleware host on and off the network.
- Changing Between a Static IP Address and DHCP
  You can change between a static IP address and DHCP.
- Using IPv6
  Oracle Fusion Middleware supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6.)

About Changing the Network Configuration

You can change the host name and IP address of an Oracle Fusion Middleware domain and the database that is used by Oracle Fusion Middleware.

- Overview of Changing the Network Configuration
- About the chghost Utility
  With the chghost utility, you can change the host name, network domain name, or IP address of a machine that contains Oracle Fusion Middleware installations or the database that contains the Oracle Fusion Middleware schemas.
- Changing the Host Name of Oracle Fusion Middleware
  You can change the host name of the host containing Oracle Fusion Middleware.
- Moving Oracle Fusion Middleware to a New Host
- Moving a Multinode Oracle Fusion Middleware to New Hosts
  If you have Oracle Fusion Middleware installed on more than one host, you can move it to new hosts.
- Changing the Host Name or IP Address of a Database
  You can change the host name or IP address of the host that contains the database that holds Oracle Fusion Middleware schema.
- Moving an Oracle Fusion Middleware Database to a New Host
  You can move the database that holds Oracle Fusion Middleware schema to a new host.
- Moving a Multinode Oracle Fusion Middleware and Its Database to New Hosts
  If you have Oracle Fusion Middleware and its database installed on more than one host, you can move it to new hosts.

- Additional Tasks for Changing the Network Configuration
  Some components require additional steps after you have run the chghost utility to change the network configuration.

Overview of Changing the Network Configuration

Using a combination of binary cloning, the chghost utility, and copying files from one host to another, Oracle Fusion Middleware supports the following changes:

- Changing the host name or IP address of the Administration Server or one or more of the Managed Servers. See Changing the Host Name of Oracle Fusion Middleware.
- Moving Oracle Fusion Middleware to a different host. See Moving Oracle Fusion Middleware to a New Host or Moving a Multinode Oracle Fusion Middleware to New Hosts.
- Changing the host name or IP address of the host that contains the database that contains the Oracle Fusion Middleware schemas. See Changing the Host Name or IP Address of a Database.
- Moving the database that contains the Oracle Fusion Middleware schemas to a different host. See Moving an Oracle Fusion Middleware Database to a New Host.
- Moving Oracle Fusion Middleware and its database to a different host. See Moving a Multinode Oracle Fusion Middleware and Its Database to New Hosts.

Note the following:

- You cannot change the topology. For example, you cannot add or remove a Managed Server and you cannot change the WebLogic Server domain name. You cannot change the port number.
- The paths of both the source and target instances must be the same. You cannot change them.
- You cannot move Oracle Fusion Middleware from one type of operating system to another. You can only move it to the same type of operating system.
- If you are moving the database from one host to another, the database type cannot be changed. For example, you cannot change the database type from an Oracle Database to a SQL Server database.
- In a multinode environment, you can move or change the network configuration of each node separately. For example, if the Administration Server is on Host A and the Managed Servers are on Host B, you can move or change the network configuration of the Administration Servers, the Managed Servers, or both.
- The chghost utility does not move binary data or data in a database.
- The chghost utility does not support environments where SSL only is configured. SSL-only means that either the administration port is enabled or only the SSL port is enabled.

The following components do not support the chghost script:

- Oracle WebCenter Sites
- Oracle BI EE
- Oracle Data Integrator
- Oracle Access Manager
- Oracle Internet Directory

The following component requires additional steps after you run the chghost utility:

- Oracle Forms Services: See Additional Tasks for Changing the Network Configuration of Oracle Forms Services.

About the chghost Utility

With the chghost utility, you can change the host name, network domain name, or IP address of a machine that contains Oracle Fusion Middleware installations or the database that contains the Oracle Fusion Middleware schemas.

The chghost utility changes any references to the host name, network domain name, or IP address within Oracle Fusion Middleware, using the information you provide in the command line or an input file.

The location of the chghost command is:

```
(UNIX) ORACLE_HOME/oracle_common/bin/chghost.sh
(Windows) ORACLE_HOME\oracle_common\bin\chghost.bat
```

If you are changing the host name, network domain name, or IP address of the Administration Server, the format of the command is:

```
./chghost.sh -chgHostInputFile ini_file
     -javaHome location_of_javahome
     -domainLoc target_domain_path
     -domainAdminUserName username
     -walletDir location_of_wallet_dir
     [-logPriority log_level]
     [-logDir location_of_log_file]
     [-ignoreValidationErrors option[, option]]
```

In a multinode environment, if you are changing the host name, network domain name, or IP address of a Managed Server, the format of the command is:

```
./chghost.sh -chgHostInputFile ini_file
     -javaHome location_of_javahome
     -domainLoc target_domain_path
     -domainAdminUserName username
     -walletDir location_of_wallet_dir
     [-logPriority log_level]
     [-logDir location_of_log_file]
     [-ignoreValidationErrors option[, option]]
     -adminURL URL_of_admin_server
     -managed
```

Table 15-1 describes the options for the chghost command.

Table 15-1 Options for the chghost Command

Option	Description
-javaHome	The Java home. The utility invokes the Java instance using the following precedence: • The value set in the command-line argument. • The value of the JAVA_HOME set in the environment. • The JAVA_HOME found in the *DOMAIN_HOME* or *ORACLE_HOME*.

Table 15-1 (Cont.) Options for the chghost Command

Option	Description
-chgHostInputFile	The absolute path, including the file name, to the input file which contains information about the server host mapping and the database host mapping. It can also contain the command line options, except for chgHostInput file.
-domainLoc	The absolute path to the domain for which you want to change the network configuration.
-domainAdminUserName	The administration user.
-walletDir	The absolute path to a directory where the chghost utility will create a file containing the administrative user password.
-logPriority	Optional. The level of the information written to the log files. Valid values are (from highest to lowest value): SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST.
-logDir	Optional. The absolute path to the directory in which the chghost utility writes log information.
-ignoreValidationErrors	Optional. The following comma-separated values are allowed : • hostname: Ignore host name validation errors. • datasource: Ignore data source validation errors. • all: Ignore both host name and data source validation errors. Note : If `ignoreValidationErrors` is specified without any value then it defaults to `all`.
-adminURL	The URL of the administration server. Use this only when you are changing the host name of a host that contains Managed Servers, not the Administration Server.
-managed	The URL of the administration server. Use this only when you are changing the host name of a host that contains Managed Servers, not the Administration Server.

You must create an input file to be passed to the chgHostInput parameter. The input file has the following format:

```
# Headers are within the brackets[]. Don't change the header name.

[ARGUMENTS]
-domainLoc domain_location
-domainAdminUserName admin_username
-walletDir wallet_location
-logPriority log_priority
-logDir log_directory
-ignoreValidationErrors option[, option]
# The following two arguments are used only when changing
# the host name of a host that contains Managed Servers,
# not the Administration Server.
-adminURL URL_of_admin_server
-managed

[SERVER_HOST_MAPPING]
#pattern source_host_name=target_host_name
source_admin_server_host.example.com=target_admin_server_host.example.com
```

```
source_admin_server_IPAddress=target_admin_server_IPAddress

source_managed_server_host.example.com=target_managed_server_host.example.com

[DATABASE_MAPPING]
#  You can only change the database host name.
db1_host.example.com=db2_host.example.com

[LDAP_MAPPING]
#  You can only change the LDAP host name.
ldap1_host.example.com=ldap2_host.example.com
```

Note the following:

- Do not change the Header names (the names within the brackets, such as [DATABASE_MAPPING]).
- For both the SERVER_HOST_MAPPING and the DATABASE_MAPPING, if you specify the full canonical name for the source system, you must specify the full canonical name for the target system. If you specify the short name for the source system, you must specify the short name for the target system. For example, the following mappings are not valid:

  ```
  host_a.example.com=host_a1.com
  host_a.com=host_a1.example.com
  ```

- The Administration Server and all Managed Servers must be stopped when you run the chghost script.
- If a different database is used for the source and target, the database port number and SID must be same on the target as on the source.
- If the chghost utility failed during execution because of a port validation error (that is, more than one server has the same port), set the following properties, then run the chghost utility again:

  ```
  -Dchghost.ignore.validation.port=true
  -chghost.temporary.port.range=lowerPort-higherport
  ```

  During execution of the chghost utility, it assigns a temporary port. Then, at the end of the execution, it will restore the original port.

  For example:
  Linux:

  ```
  CHGHOST_JAVA_OPTIONS="$CHGHOST_JAVA_OPTIONS
  -Dchghost.ignore.validation.port=true
  -Dchghost.temporary.port.range=443-10000"
  export CHGHOST_JAVA_OPTIONS
  ```

  Windows:

  ```
  set CHGHOST_JAVA_OPTIONS]=%CHGHOST_JAVA_OPTIONS%
  -Dchghost.ignore.validation.port=true
  -Dchghost.temporary.port.range=443-10000
  ```

Changing the Host Name of Oracle Fusion Middleware

You can change the host name of the host containing Oracle Fusion Middleware.

In this scenario, you have an Administration Server on Host_A, Managed Servers on Host_B, and the database on Host_C. You have changed the host name of one or more of these hosts. In the following example, you want to change Host_A to Host_A1 and Host_B to Host_B1.

To change the host name for the Administration Server from Host_A to Host_A1:

1. Change the hostname, domain name, or IP address of your host.
2. Stop the Administration Server and all Managed Servers if they are running.
3. Create an input file that contains the following:

   ```
   [ARGUMENTS]

   [SERVER_HOST_MAPPING]
   #pattern source_host_name=target_host_name
   Host_A.domain.com=Host_A1.domain.com
   Host_B.domain.com=Host_B1.domain.com
   ```

   You can pass any of the arguments, except chgHostInputFile, in the input file. In this example, you pass the arguments on the command line.

   If all of the servers are on the same host, you do not need to take any separate action for them.
4. Create a wallet directory, specifying the file specification, including the name of the directory and passing the administration user name to the command. For example:

   ```
   ORACLE_HOME/oracle_common/common/bin/configWallet.sh -walletDir /scratch/
   myWalletDir admin_username
   ```

   The command will prompt you to enter the password and to confirm the password. It creates the wallet directory if it does not exist and populates it with a cwallet.sso file.
5. Execute the following command on Host_A1:

   ```
   ./chghost.sh -chgHostInputFile input_file
        -javaHome location_of_javahome
        -domainLoc target_domain_path
        -domainAdminUserName username
        -walletDir location_of_wallet_dir
        -logPriority log_level
        -logDir location_of_log_file
   ```

6. To change the name of Host_B, which holds the Managed Servers, to Host_B1, create an input file that contains the following:

   ```
   [ARGUMENTS]

   [SERVER_HOST_MAPPING]
   #pattern source_host_name=target_host_name
   ```

```
Host_A.domain.com=Host_A1.domain.com
Host_B.domain.com=Host_B1.domain.com
```

7. Create a wallet file, as described in Step 4.
8. Execute the chghost command on Host_B1, as described in Step 5.

The references within Oracle Fusion Middleware are changed to the new host names.

Moving Oracle Fusion Middleware to a New Host

In this scenario, you want to move Oracle Fusion Middleware to a new host. It is currently on Host_A and you want to move it to Host_A1.

You use the copyBinary and pasteBinary commands to move the binary files and you use the chghost command to change the host name or IP address of Oracle Fusion Middleware.

1. On Host_A, execute the copyBinary script:

```
cd ORACLE_HOME/oracle_common/bin/
./copyBinary.sh -javaHome path_of_jdk
                -sourceOracleHomeLoc OracleHome_location
                -archiveLoc location_of_resulting_jar_file
```

   See copyBinary Script for the complete syntax of the copyBinary script.
2. Copy the resulting jar file to Host_A1.
3. Unpack the jar file. Use the pasteBinary script on Host_A1. The script is located in:

```
(UNIX) ORACLE_HOME/oracle_common/bin/pasteBinary.sh
(Windows) ORACLE_HOME\oracle_common\bin\pasteBinary.cmd
```

   See pasteBinary Script for the complete syntax.

   If Oracle Fusion Middleware is not installed on Host_A1 or Oracle Home does not exist on Host_A1, follow these steps:

   (Unix)

   a. Create a new Oracle_Home directory on Host_A1.
   b. Copy `oraInst.loc` from *`sourceOracleHomeLoc`* to *`targetOracleHomeLoc`* on Host_A1.
   c. Verify the following values of *`targetOracleHomeLoc`*`/oraInst.loc`. Modify them if required.
      - `inst_group=`*`dbs`* where *`dbs`* is a sample value for the operating system group whose members can write to the Oracle Inventory (oraInventory) and the user is a member of that group.
      - `inventory_loc=`*`targetOracleHomeLoc`*`/oraInventory`
   d. Run the following command:

```
/scratch/jdk17.0.12_131/bin/java -jar /scratch/archive.jar
            -targetOracleHomeLoc /scratch/Oracle_home
            -invPtrLoc /scratch/oracle/oraInst.loc
            -javaHome /scratch/jdk17.0.12_131/
```

(Windows)

- Run the following command:

```
\scratch\jdk17.0.12_131\bin\java -jar \scratch\archive.jar
            -targetOracleHomeLoc \scratch\Oracle_home
            -javaHome \scratch\jdk17.0.12_131\
```

4. Copy the domain home to Host_A1.
   a. Use a utility to create an archive of the domain home on Host_A. For example, use zip to archive the domain home directory structure for the domain soa_domain, on Host_A:

      ```
      cd ORACLE_HOME/domains
      zip domain.zip -r soa_domain
      ```

      You can also use the tar command.

   b. Copy the archive to Host_A1 and use unzip to unpack the archive. You must unpack it in the same directory structure as on Host A. For example, if the domain was in *ORACLE_HOME*/domains on Host_A, it must be in *ORACLE_HOME*/domains on Host_A1.

      ```
      cd ORACLE_HOME/domains
      unzip domain.zip
      ```

5. On Host_A1, create an input file that contains the following:

   ```
   [ARGUMENTS]

   [SERVER_HOST_MAPPING]
   #pattern source_host_name=target_host_name
   Host_A.example.com=Host_A1.example.com
   ```

6. On Host_A1, create a wallet directory, specifying the full path, including the name of the directory and passing the administration user name to the command. For example:

   ```
   ORACLE_HOME/oracle_common/common/bin/configWallet.sh -walletDir /scratch/
   myWalletDir admin_username
   ```

   The command will prompt you to enter the password and to confirm the password. It creates the wallet directory if it does not exist and populates it with a cwallet.sso file.

7. Run the chghost command on Host_A1:

   ```
   ./chghost.sh -chgHostInputFile input_file
        -javaHome location_of_javahome
        -domainLoc target_domain_path
        -domainAdminUserName username
        -walletDir location_of_wallet_dir
        -logPriority log_level
        -logDir location_of_log_file
   ```

Now, Oracle Fusion Middleware is on Host_A1 and the host name has been changed in the Oracle Fusion Middleware instance.

Moving a Multinode Oracle Fusion Middleware to New Hosts

If you have Oracle Fusion Middleware installed on more than one host, you can move it to new hosts.

In this scenario, you have Oracle Fusion Middleware installed on more than one host and you want to move to new hosts. The Administration Server is in Host_A and the Managed Servers are on Host_B You want to move the Admnistration Server to Host_A1 and the Managed Servers to Host_B1.

You use the copyBinary and pasteBinary commands to move the binary files and you use the chghost command to change the host name or IP address of Oracle Fusion Middleware.

1. On Host_A, execute the copyBinary script:

   ```
   cd ORACLE_HOME/oracle_common/bin/
   ./copyBinary.sh -javaHome path_of_jdk
                   -sourceOracleHomeLoc OracleHome_location
                   -archiveLoc location_of_resulting_jar_file
   ```

   See copyBinary Script for the complete syntax of the copyBinary script.
2. Copy the resulting jar file to Host_A1.
3. Unpack the jar file:
   - Use the pasteBinary script on Host_A1. The script is located in:

     ```
     (UNIX) ORACLE_HOME/oracle_common/bin/pasteBinary.sh
     (Windows) ORACLE_HOME\oracle_common\bin\pasteBinary.cmd
     ```

     If Oracle Fusion Middleware is not installed on Host_A1, copy the following files from Host_A:

     ```
     (UNIX) ORACLE_HOME/oracle_common/bin/pasteBinary.sh
     (Windows) ORACLE_HOME\oracle_common\bin\pasteBinary.cmd
     ```

     Execute the following command:

     ```
     pasteBinary.sh -javaHome /scratch/oracle/jdk17.0.12_40
                    -archiveLoc  /tmp/oh_copy.jar
                    -targetOracleHomeLoc /scratch/oracle/Oracle_home_prod
                    -targetOracleHomeName Oracle_home_prod
     ```

     See pasteBinary Script for the complete syntax.
   - If an Oracle Home does not exist on Host_A1, use the following command:

     ```
     /scratch/jdk17.0.12_131/bin/java -jar /scratch/archive.jar
           -targetOracleHomeLoc /scratch/Oracle_home
           -invPtrLoc /scratch/oracle/oraInst.loc
           -javaHome /scratch/jdk17.0.12_131/
     ```
4. Copy the domain home to Host_A1.

a. Use a utility to create an archive of the domain home on Host_A. For example, use zip to archive the domain home directory structure for the domain soa_domain, on Host_A:

```
cd ORACLE_HOME/domains
zip domain.zip -r soa_domain
```

b. Copy the archive to Host_A1 and use unzip to unpack the archive. You must unpack it in the same directory structure as on Host_A. For example, if the domain was in *ORACLE_HOME*/domains on Host_A, it must be in *ORACLE_HOME*/domains on Host_A1.

```
cd ORACLE_HOME/domains
unzip domain.zip
```

5. Create an input file that contains the following:

```
[ARGUMENTS]

[SERVER_HOST_MAPPING]
#pattern source_host_name=target_host_name
Host_A.domain.com=Host_A1.domain.com
Host_B.domain.com=Host_B1.domain.com
```

6. Create a wallet directory, specifying the full path, including the name of the directory and passing the administration user name to the command. For example:

```
ORACLE_HOME/oracle_common/common/bin/configWallet.sh -walletDir /scratch/
myWalletDir admin_username
```

The command will prompt you to enter the password and to confirm the password. It creates the wallet directory if it does not exist and populates it with a cwallet.sso file.

7. Run the chghost command on Host_A1:

```
./chghost.sh -chgHostInputFile input_file
     -javaHome location_of_javahome
     -domainLoc target_domain_path
     -domainAdminUserName username
     -walletDir location_of_wallet_dir
     -logPriority log_level
     -logDir location_of_log_file
```

8. On Host_B, execute the copyBinary script:

```
cd ORACLE_HOME/oracle_common/bin/
./copyBinary.sh -javaHome path_of_jdk
                -sourceOracleHomeLoc OracleHome_location
                -archiveLoc location_of_resulting_jar_file
```

See copyBinary Script for the complete syntax of the copyBinary script.

9. Copy the resulting jar file to Host_B1.

10. Unpack the jar file. If an Oracle Home exists on Host_B1, you can use the pasteBinary script, as described in pasteBinary Script . If an Oracle Home does not exist on Host_B1, use the following command:

```
/scratch/jdk17.0.12_131/bin/java -jar /scratch/archive.jar
   -targetOracleHomeLoc /scratch/Oracle_home
   -invPtrLoc /scratch/oracle/oraInst.loc
   -javaHome /scratch/jdk17.0.12_131/
```

11. Copy the domain home to Host_B1.

    a. Use a utility to create an archive of the domain home on Host_B. For example, use zip to archive the domain home directory structure for the domain soa_domain, on Host_B:

    ```
    cd ORACLE_HOME/domains
    zip domain.zip -r soa_domain
    ```

    b. Copy the archive to Host_B1 and use unzip to unpack the archive. You must unpack it in the same directory structure as on Host_B. For example, if the domain was in *ORACLE_HOME*/domains on Host_B, it must be in *ORACLE_HOME*/domains on Host_B1.

    ```
    cd ORACLE_HOME/domains
    unzip domain.zip
    ```

12. On Host_B1, create an input file that contains the following:

```
[ARGUMENTS]

[SERVER_HOST_MAPPING]
#pattern source_host_name=target_host_name
Host_A.domain.com=Host_A1.domain.com
Host_B.domain.com=Host_B1.domain.com
```

13. Create a wallet directory, as described in Step 6.
14. Run the chghost command on Host_B1, passing it the `adminURL` and `managed` options:

```
./chghost.sh -chgHostInputFile input_file
     -javaHome location_of_javahome
     -domainLoc target_domain_path
     -domainAdminUserName username
     -walletDir location_of_wallet_dir
     -logPriority log_level
     -logDir location_of_log_file
     -adminURL t3://target_admin_host:target_domain_port
     -managed
```

The references within Oracle Fusion Middleware are changed to the new host names.

Changing the Host Name or IP Address of a Database

You can change the host name or IP address of the host that contains the database that holds Oracle Fusion Middleware schema.

In this scenario, you have a database on Host_C and you have changed the name of the host to Host_C1. The Adminstration Server is on Host_A and the Managed Servers are on Host_B. You are not changing the names or moving the Adminstration Server or Managed Servers. To change the host name for the database from Host_C to Host_C1:

1. Change the hostname, domain name, or IP address of your host.
2. Create an input file that contains the following:

```
[ARGUMENTS]

[DATABASE_MAPPING]
#  You can only change the database host name.
Host_C.domain.com=Host_C1.domain.com
```

   You can pass any of the arguments, except chgHostInputFile, in the input file. In this example, you pass the arguments on the command line.
3. Stop the Administration Server and the Managed Servers.
4. Execute the following command on the Administration Server host and on the Managed Server host:

```
./chghost.sh -chgHostInputFile input_file
     -javaHome location_of_javahome
     -domainLoc target_domain_path
     -domainAdminUserName username
     -walletDir location_of_wallet_dir
     -logPriority log_level
     -logDir location_of_log_file
```

   The `chghost` command makes the necessary changes. It automatically start the servers in the domain on the current machine, and then may shut them down. In some cases, the shutdown attempt might fail.
5. Start the Administration Server and the Managed Servers.

> **Note:**
>
> If the `chghost` command shuts down the Administration Server and the Managed Servers, you should start them manually.

The references within Oracle Fusion Middleware are changed to the new host name.

Moving an Oracle Fusion Middleware Database to a New Host

You can move the database that holds Oracle Fusion Middleware schema to a new host.

In this scenario, you have a database on Host_C and you want to move it to Host_C1. The Adminstration Server is on Host_A and the Managed Servers are on Host_B. You are not changing the names or moving the Adminstration Server or Managed Servers.

1. Move the database to a new host, using Oracle Database methods, such as RMAN or the Oracle Database export and import utilities. For more information, see Fusion Middleware Creating Schemas with the Repository Creation Utility.

   Note the following:

   - The service name and SID of the database on the new host must be the same as on the source host.
   - The port on the new host must be the same as on the source host.
   - The database on the new host must contain the same schemas as on the source host and the schemas must have the same prefix as on the source host.

2. Create an input file that contains the following:

   ```
   [ARGUMENTS]

   [DATABASE_MAPPING]
   #  You can only change the database host name.
   Host_C.domain.com=Host_C1.domain.com
   ```

   You can pass any of the arguments, except chgHostInputFile, in the input file. In this example, you pass the arguments on the command line.

3. Stop the Administration Server and the Managed Servers.

4. Execute the following command on the Administration Server host (HOST_A) and on the Managed Server host (HOST_B):

   ```
   ./chghost.sh -chgHostInputFile input_file
        -javaHome location_of_javahome
        -domainLoc target_domain_path
        -domainAdminUserName username
        -walletDir location_of_wallet_dir
        -logPriority log_level
        -logDir location_of_log_file
   ```

5. Start the Administration Server and the Managed Servers.

Moving a Multinode Oracle Fusion Middleware and Its Database to New Hosts

If you have Oracle Fusion Middleware and its database installed on more than one host, you can move it to new hosts.

In this scenario, you have Oracle Fusion Middleware installed on more than one host and you want to move to new hosts. The Administration Server is in Host_A and the Managed Servers are on Host_B You want to move the Admnistration Server to Host_A1 and the Managed

Servers to Host_B1. In addition, you have the database installed on Host_C and you want to move it to Host_C1.

You use the copyBinary and pasteBinary commands to move the binary files and you use the chghost command to change the host name or IP address of Oracle Fusion Middleware.

1. Create an input file that contains the following:

```
[ARGUMENTS]

[SERVER_HOST_MAPPING]
#pattern source_host_name=target_host_name
Host_A.domain.com=Host_A1.domain.com
Host_B.domain.com=Host_B1.domain.com


[DATABASE_MAPPING]
Host_C.domain.com=Host_C1.domain.com
```

   You will use this file when you run the chghost command.

2. Run the copyBinary script on Host_A, copy the jar file to Host_A1, run the pasteBinary command on Host_A1 and copy the domain from Host_A to Host_A1, as described in Steps 1 through 4 in Moving a Multinode Oracle Fusion Middleware to New Hosts.
3. Create a wallet directory, as described in Step 6 in Moving a Multinode Oracle Fusion Middleware to New Hosts.
4. Run the chghost command on Host_A1, as described in Step 7 in Moving a Multinode Oracle Fusion Middleware to New Hosts. Use the input file shown in the first step.
5. Run the copyBinary script on Host_B, copy the jar file to Host_B1, run the pasteBinary command on Host_B1 and copy the domain from Host_B to Host_B1, as described in Steps 8 through 11 in Moving a Multinode Oracle Fusion Middleware to New Hosts.
6. Create a wallet directory, as described in Step 6 in Moving a Multinode Oracle Fusion Middleware to New Hosts.
7. Run the chghost command on Host_B1, passing it the `adminURL` and `managed` options, as described in Step 14 in Moving a Multinode Oracle Fusion Middleware to New Hosts.
8. Move the database to a new host, using Oracle Database methods. See Step 1 in Moving an Oracle Fusion Middleware Database to a New Host.
9. Create an input file that contains the following:

```
[ARGUMENTS]

[DATABASE_MAPPING]
#  You can only change the database host name.
Host_C.domain.com=Host_C1.domain.com
```

10. Stop the Administration Server and the Managed Servers.
11. Execute the chghost command on Administration Server host (HOST_A1) and on the Managed Server host (HOST_B1), as described in Step 4 in Moving an Oracle Fusion Middleware Database to a New Host.
12. Start the Administration Server and the Managed Servers.

Additional Tasks for Changing the Network Configuration

Some components require additional steps after you have run the chghost utility to change the network configuration.

- Additional Tasks for Changing the Network Configuration of Oracle Forms Services
  After you run the chghost utility, you need to take additional steps to update the host and port information for Forms Application Deployment Services.

Additional Tasks for Changing the Network Configuration of Oracle Forms Services

After you run the chghost utility, you need to take additional steps to update the host and port information for Forms Application Deployment Services.

To update the host name and port information, you run the following script:

```
ORACLE_HOME/forms/fads/fads_config.py
```

The script takes the following mandatory arguments:

- The Administration Server host name
- The Administration Server port
- The applications directory

For example:

```
wlst.sh ORACLE_HOME/forms/fads/fads_config.py updateHostPort myserverhost myserverport
Oracle_Home/applications/forms_domain
```

Moving Between On-Network and Off-Network

You can move an Oracle Fusion Middleware host on and off the network.

The following assumptions and restrictions apply:

- The host must contain an instance that does not use an Infrastructure, or both the middle-tier instance and Infrastructure must be on the same host.
- DHCP must be used in loopback mode. Refer to the *Oracle Fusion Middleware System Requirements and Specifications*.
- 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).

This section contains the following topics:

- Moving from Off-Network to On-Network (Static IP Address)
- Moving from Off-Network to On-Network (DHCP)
- Moving from On-Network to Off-Network (Static IP Address)

Moving from Off-Network to On-Network (Static IP Address)

This procedure assumes you have installed Oracle Fusion Middleware on a host that is off the network, using a standard host name (not `localhost`), and would like to move on to the network and use a static IP address. The IP address may be the default loopback IP, or any standard IP address.

To move on to the network, you can simply connect the host to the network. No updates to Oracle Fusion Middleware are required.

Moving from Off-Network to On-Network (DHCP)

This procedure assumes you have installed on a host that is off the network, using a standard host name (not `localhost`), and would like to move on to the network and use DHCP. The IP address of the host can be any static IP address or loopback IP address, and should be configured to the host name.

To move on to the network:

1. Connect the host to the network using DHCP.
2. Configure the host name to the loopback IP address only.

Moving from On-Network to Off-Network (Static IP Address)

Follow this procedure if your host is on the network, using a static IP address, and you would like to move it off the network:

1. Configure the `/etc/hosts` file so the IP address and host name can be resolved locally.
2. Take the host off the network.

   There is no need to perform any steps to change the host name or IP address.

Changing Between a Static IP Address and DHCP

You can 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 the *Oracle Fusion Middleware System Requirements and Specifications*.
- 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.
- Changing from a Static IP Address to DHCP
- Changing from DHCP to a Static IP Address

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.

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.

Using IPv6

Oracle Fusion Middleware supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6.)

Among other features, IPv6 supports a larger address space (128 bits) than IPv4 (32 bits), providing an exponential increase in the number of computers that can be addressable on the Web.

An IPv6 address is expressed as 8 groups of 4 hexadecimal digits. For example:

```
2001:0db8:85a3:08d3:1319:8a2e:0370:7334
```

For information about the support for IPv6 by Oracle Fusion Middleware components, see *Oracle Fusion Middleware System Requirements and Specifications*.

Most components support IPv6. The following topics provide more information about configuring Oracle Fusion Middleware certain components for IPv6:

- Configuring Oracle HTTP Server for IPv6
- Using Dual Stack with Oracle SOA Suite, Oracle Identity Governance, and Fusion Middleware Control

Configuring Oracle HTTP Server for IPv6

To configure Oracle HTTP Server to communicate using IPv6, you modify configuration files in the following directory:

```
(UNIX) DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_name
(Windows) DOMAIN_HOME\config\fmwconfig\components\OHS\ohs_name
```

For example, to configure Oracle HTTP Server to communicate with Oracle WebLogic Server on hosts that are running IPv6, you configure mod_wl_ohs. You edit the configuration files in the following directory:

```
DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs1
```

In the files, specify either the resolvable host name or the IPv6 address in one of the following parameters:

```
WebLogicHost hostname | [IPaddress]
WebCluster [IPaddress_1]:portnum1, [IPaddress_2]:portnum2, [IPaddress_3]:portnum3, ...
```

You must enclose the IPv6 address in brackets.

Any errors are logged in the Oracle HTTP Server logs. To generate more information, set the mod_weblogic directives Debug All and WLLogFile path. Oracle HTTP Server logs module-specific messages.

> **Note:**
>
> In previous versions, Oracle HTTP Server contained restrictions about using dynamic clusters with IPv6 nodes. For example, the Oracle HTTP Server plug-in for Oracle WebLogic Server had limited IPv6 support in that the DSL (dynamic server list) feature of the plug-in was not supported; only the static configuration of server lists was supported (DynamicServerList=OFF). Those restrictions have been lifted.
>
> The OHS installation/configuration comes with a default SSL wallet containing a certificate with subject line having localhost mentioned in it.
>
> If you put the localhost in `admin.conf` in `Listen`, `VirtualHost`, and `ServerName` directives then everything works fine.
>
> If you don't want to use the localhost in `admin.conf` then, you can use the `fqdn` name of that host.
>
> For example:
>
> `abc.xyz.com` in `Listen`, `VirtualHost`, and `ServerName` directives. Here you have to create a new wallet with a new certificate containing `abc.xyz.com` in the certificate subject line and should specify this new wallet in the `admin.conf`.
>
> The nodemanager OHS plugin reads the OHS `admin.conf` and uses these settings to connect to OHS.

Using Dual Stack with Oracle SOA Suite, Oracle Identity Governance, and Fusion Middleware Control

Oracle SOA Suite and Oracle Identity Governance supports a dual-stack configuration. However, when you use Fusion Middleware Control with Oracle SOA Suite or Oracle Identity Governance, you must specify the protocol in the following file. Otherwise, Fusion Middleware Control may not work correctly.

DSOMAIN_HOME`/bin/startWebLogic.sh`

In the file, add the following line, specifying the IP protocol after the line `${DOMAIN_HOME}/bin/setDomainEnv.sh`:

```
$DOMAIN_HOME/bin  JAVA_OPTIONS="${JAVA_OPTIONS} -Djava.net.preferIPv4Stack=true"
```

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.

16

Introduction to 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. Oracle Fusion Middleware provides recommendations for backup and recovery.

- About Oracle Fusion Middleware Backup and Recovery
  It is important to consider your entire Oracle Fusion Middleware environment when performing backup and recovery.
- Oracle Fusion Middleware Directory Structure
  Oracle Fusion Middleware has a standard directory structure, which is important to understand when you are backing up or recovering Oracle Fusion Middleware.
- Tools to Use for Backup and Recovery
  You can use standard operating system tools to backup or recover Oracle Fusion Middleware.
- Backup and Recovery Recommendations for Oracle Fusion Middleware Components
  Oracle Fusion Middleware components have different requirements for what you back up and what you recover. Many components have dependencies on a database or on other components.
- Assumptions and Restrictions for Backup and Recovery
  Certain assumptions and restrictions apply to the backup and recovery procedures in this book.

About Oracle Fusion Middleware Backup and Recovery

It is important to consider your entire Oracle Fusion Middleware environment when performing 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 a WebLogic Server domain with Identity Management components.

The installations of an Oracle Fusion Middleware environment are interdependent in that they contain configuration information, applications, and data that are kept in synchronization. For example, when you perform a configuration change, information in configuration files is updated. When you deploy an application, you might deploy it to all Managed Servers in a domain or cluster.

It is, therefore, important to consider your entire Oracle Fusion Middleware environment when performing backup and recovery. You should back up your entire Oracle Fusion Middleware environment at once, then periodically. If a loss occurs, you can restore your environment to a consistent state.

> **See Also:**
>
> *Understanding Oracle Fusion Middleware* for conceptual information about Oracle WebLogic Server domains, the Administration Server, Managed Servers and clusters, and Node Manager.

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

Impact of Administration Server Failure

The failure of an Administration Server does not affect the operation of Managed Servers in the domain but it does prevent you from changing the domain's configuration. If an Administration Server fails because of a hardware or software failure on its host computer, other server instances on the same computer may be similarly affected.

If an Administration Server for a domain becomes unavailable while the server instances it manages—clustered or otherwise—are running, those Managed Servers continue to run. Periodically, these Managed Servers attempt to reconnect to the Administration Server. For clustered Managed Server instances, the load balancing and failover capabilities supported by the domain configuration continue to remain available.

When you first start a Managed Server, it must be able to connect to the Administration Server to retrieve a copy of the configuration. Subsequently, you can start a Managed Server even if the Administration Server is not running. In this case, the Managed Server uses a local copy of the domain's configuration files for its starting configuration and then periodically attempts to connect with the Administration Server. When it does connect, it synchronizes its configuration state with that of the Administration Server.

Managed Server Independence (MSI) Mode

A Managed Server maintains a local copy of the domain configuration. When a Managed Server starts, it contacts its Administration Server to retrieve any changes to the domain configuration that were made since the Managed Server was last shut down. If a Managed Server cannot connect to the Administration Server during startup, it can use its locally cached configuration information—this is the configuration that was current at the time of the Managed Server's most recent shutdown. A Managed Server that starts without contacting its Administration Server to check for configuration updates is running in Managed Server Independence (MSI) mode. By default, MSI mode is enabled. However, a Managed Server cannot be started, even in MSI mode, for the first time if the Administration Server is down due to non-availability of the cached configuration.

Configuration Changes in Managed Servers

Configuration changes are updated in a Managed Server during the following events:

- On each Managed Server restart, the latest configuration is retrieved from the Administration Server. This happens even when Node Manager is down on the node where the Managed Server is running. If the Administration Server is unavailable during

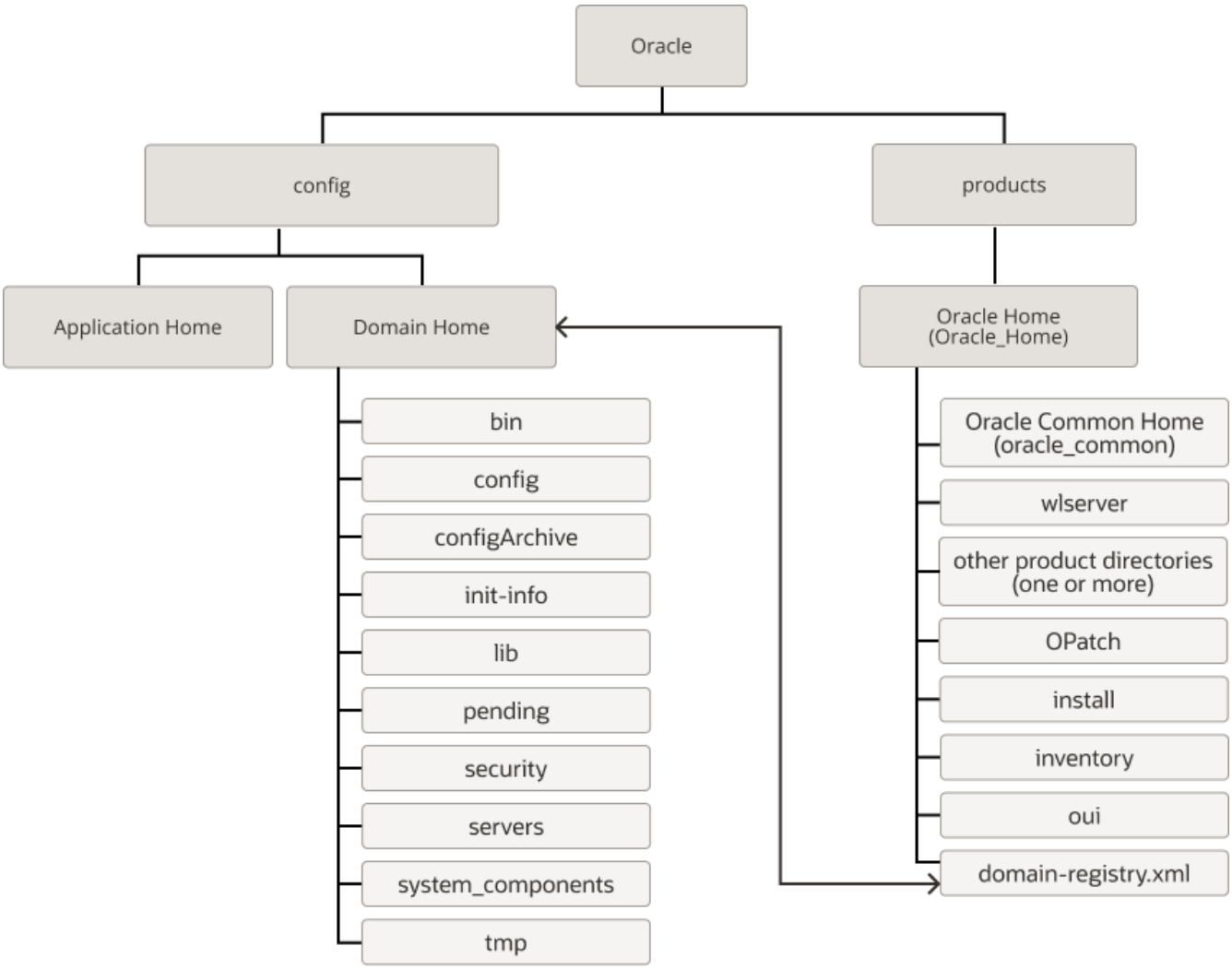
the Managed Server restart and if the MSI (Managed Server Independence) mode is enabled in the Managed Server, it starts by reading its local copy of the configuration and synchronizes with the Administration Server when it is available. By default MSI mode is enabled.

- Upon activating every administrative change such as configuration changes, deployment or redeployment of applications, and topology changes, the Administration Server pushes the latest configuration to the Managed Server. If the Managed Server is not running, the Administration Server pushes the latest version of the configuration to the Managed Server when it does start.

Oracle Fusion Middleware Directory Structure

Oracle Fusion Middleware has a standard directory structure, which is important to understand when you are backing up or recovering Oracle Fusion Middleware.

The following shows a simplified view of the Oracle Fusion Middleware directory structure when you have installed the Oracle Fusion Middleware Infrastructure:



Tools to Use for Backup and Recovery

You can use standard operating system tools to backup or recover Oracle Fusion Middleware.

To backup or recover your Oracle Fusion Middleware environment, you can use:

- File copy utilities such as copy, xcopy, tar, or jar. Make sure that the utilities:
  - Preserve symbolic links
  - Support long file names
  - Preserve the permissions, timestamps, and ownership of the files

  When you backup and restore the files, use your preferred tool. For example:

  - On Windows, for online backup and recovery, use copy or xcopy; for offline backup and recovery, use copy, xcopy, or jar. Do not use Winzip because it does not work with long filenames or extensions.

    Note that for some versions of Windows, any file name with more than 256 characters fails. You can use the xcopy command with the following switches to work around this issue:

    ```
    xcopy /s/e  "C:\Temp\*.*"  "C:\copy"
    ```

    See the xcopy help for more information about syntax and restrictions.
  - On Linux and UNIX, use tar.

  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.
- Oracle Recovery Manager (RMAN) to back up or recover database-based metadata repositories and any databases used by Oracle Fusion Middleware. With RMAN, you can perform full backups or incremental backups. See *Oracle Database Backup and Recovery User's Guide* for information about using RMAN to back up or recovery a database.

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.

Backup and Recovery Recommendations for Oracle Fusion Middleware Components

Oracle Fusion Middleware components have different requirements for what you back up and what you recover. Many components have dependencies on a database or on other components.

Table 16-1 describes what you must back up and recover for each Oracle Fusion Middleware component. Note the following:

- If the component has a database dependency listed in the table, back up and recover the database using RMAN, as described in the *Oracle Database Backup and Recovery User's Guide* .
- For backup, back up the entity listed in the table, as described in Performing a Backup.
- For recovery, depending on what has failed, you may need to recover the following, as described in Recovering Your Environment:
  - The domain, which, for some components, can be either a WebLogic Server domain (see Recovering an Oracle WebLogic Server Domain) or a standalone domain (see Recovering a Standalone Domain.)
  - The Administration Server configuration: See Recovering the Administration Server Configuration.
  - A Managed Server: See Recovering a Managed Server.
  - A cluster: See Recovering a Cluster.
  - Applications: See Recovering Applications.
  - The database containing the schemas related to the component. See Recovering a Database.

  After a loss of host, you may need to recover the following:
  - The domain, which, for some components, can be either a WebLogic Server domain (see Recovering After Loss of Oracle WebLogic Server Domain Host) or a standalone domain (see Recovering After Loss of Standalone Domain Host).
  - The Administration Server host: See Recovering After Loss of Administration Server Host.
  - The Managed Server host: See Recovering After Loss of Managed Server Host.
  - The database containing the schemas related to the component. See Recovering After Loss of Database Host.

Table 16-1 describes the backup and recovery recommendations for Oracle Fusion Middleware components.

Table 16-1 Backup and Recovery Recommendations

Component	Database Dependencies	Backup Recommendation	Recovery Recommendation	Additional Information
Oracle Access Management Access Manager	The schema used by the Access Manager policy store.	The Middleware home and the domain home for the Access Manager server.	The Middleware home and the domain home for the Access Manager server. Also the Oracle home and the domain for the Oracle HTTP Server that contains the Webgate, as needed.	To recover from loss of host, see *Introduction to Backup and Recovery*.
Oracle B2B	MDS schema	The Administration Server domain directory, the Oracle home, and the product home if changes are made to the Oracle B2B configuration file	The Managed Server	See *Recovering Oracle B2B* for information about the file Xengine.tar.gz.

Table 16-1 (Cont.) Backup and Recovery Recommendations

Component	Database Dependencies	Backup Recommendation	Recovery Recommendation	Additional Information
Oracle BPEL Process Manager	MDS and SOAINFRA schemas	The Oracle home and the Administration Server domain directory	The domain home and the Managed Server	See Backup and Recovery Recommendations for Oracle BPEL Process Manager for information about backing up and recovering the database.
Oracle Business Activity Monitoring	MDS and SOAINFRA schemas	The Oracle home, the Administration Server domain directory, the Managed Server directory.	The Managed Server or the Oracle home, or both, depending on the extent of failure	NA
Oracle Business Process Management	MDS schema	The Administration Server domain directory and the same data as Oracle BPEL Process Manager, as described in Backup and Recovery Recommendations for Oracle BPEL Process Manager.	The same data as Oracle BPEL Process Manager and the Managed Server	NA
Oracle Business Rules	MDS schema	The Oracle home and the Administration Server domain directory	The Managed Server where the soa-infra application is deployed	NA
Oracle Data Integrator	ODI_REPO schema	The Oracle home, the domain if Oracle Data Integrator is installed in a domain, and the ODI_Oracle_Home/oracledi/agent folder for each machine where a standalone agent is installed	The Managed Server or the Oracle home, or both. If your environment contains the Oracle Data Integrator Standalone Agent or Oracle Data Integrator for Developers, restore the Oracle home, as described in Recovering the Oracle Home. If your environment contains Oracle Data Integrator deployed in a Managed Server, restore the Managed Server, as described in Recovering a Managed Server.	To recover from loss of host, see Recovering Oracle Data Integrator to a Different Host.
Oracle Enterprise Scheduler	ESS schema	The Oracle home and the Administration Server domain directory	The entity that has failed	NA

Table 16-1 (Cont.) Backup and Recovery Recommendations

Component	Database Dependencies	Backup Recommendation	Recovery Recommendation	Additional Information
Oracle Forms Services	Any user-configured database for Oracle Forms Services applications.	The Administration Server domain, the Managed Server directory, and domain directory where Oracle Forms Services is located.	The domain directory where Oracle Forms Services is located.	To recover from loss of host, see Recovering Oracle Forms Services to a Different Host
Oracle HTTP Server	None	The Oracle home and the domain, which can be either a standalone domain or the Oracle WebLogic Server domain.	The Administration Server domain directory	NA
Oracle Identity Governance	OIM, MDS, OPSS, and Oracle SOA Suite schemas and, optionally, the OID schema	The domain, the Oracle home	The domain or Oracle home depending on the extent of the failure.	To recover from loss of host, see *Recovering Oracle Identity Governance to a Different Host*.
Oracle Internet Directory	ODS and ODSSM schemas	The domain, which can be either a standalone domain or the Oracle WebLogic Server domain.	The domain, which can be either a standalone domain or the Oracle WebLogic Server domain.	To recover from loss of host, see *Recovering Oracle Internet Directory to a Different Host*.
Oracle Managed File Transfer	MFT and MDS and schemas	The Oracle home and the Administration Server domain directory	The entity that has failed	NA
Oracle Mediator	MDS and SOAINFRA schemas	The Oracle home and the Administration Server domain directory	The Managed Server where the soa-infra application is deployed	NA
Oracle Platform Security Services	If a database-based Oracle Platform Security repository is used, the OPSS schema. If an Oracle Internet Directory based repository is used, an Oracle Internet Directory repository.	The Oracle home and the Administration Server domain directory. Back up Oracle Internet Directory if Oracle Platform Security uses an Oracle Internet Directory based repository.	The files listed in Recovering Oracle Platform Security Services.	NA
Oracle Reports	The database containing any job-related information	The Administration Server domain and the Managed Server directory where Oracle Reports is located	The Administration Server domain and the Managed Server directory where Oracle Reports is located.	To recover from loss of host, see Recovering Oracle Reports to a Different Host

Table 16-1 (Cont.) Backup and Recovery Recommendations

Component	Database Dependencies	Backup Recommendation	Recovery Recommendation	Additional Information
Oracle Service Bus	If its reporting feature is enabled, Oracle Service Bus creates two tables, WLI_QS_REPORT_DATA and WLI_QS_REPORT_ATTRIBUTE, in a user-specified schema.	The Oracle home and the Administration Server domain directory	The Managed Server	NA
Oracle SOA Suite	MDS and SOAINFRA schemas	The Oracle home and the Administration Server domain directory	The entity that has failed	For loss of host, see Recovering Oracle SOA Suite After Loss of Host. See Backup and Recovery Recommendations for Oracle BPEL Process Manager for information about backing up and recovering the database.
Oracle User Messaging Service	UMS schema	The Oracle home and the domain, which can be either a standalone domain or the Oracle WebLogic Server domain.	The domain, which can be either a standalone domain or the Oracle WebLogic Server domain	Make configuration changes as described in Recovering After Loss of Standalone Domain Host or Recovering After Loss of Managed Server Host.
Oracle Web Services Manager	If a database-based MDS Repository is used, the MDS schema.	The Oracle home and the Administration Server domain directory. If Oracle WSM uses a file-based MDS repository, back it up using a file copy mechanism.	The Managed Server If Oracle WSM uses a file-based MDS repository, restore it from backup.	NA
Oracle WebCenter Capture	CAPTURE schema	The Oracle home and the Administration Server domain directory	The entity that has failed	NA

Table 16-1 (Cont.) Backup and Recovery Recommendations

Component	Database Dependencies	Backup Recommendation	Recovery Recommendation	Additional Information
Oracle WebCenter Content	OCS schemas	The domain and the Oracle home If the Vault and WebLayout directories are not located in the domain directory, back up their directories, which are specified in: *DOMAIN_HOME*/ucm/CONTEXT-ROOT/config/config.cfg Also, back up the following directory, which is located in a shared file system: *DOMAIN_HOME*/ucm/CONTEXT-ROOT/config	The domain and the shared file system containing the Vault and WebLayout directories, depending on the severity of the failure.	To recover, see Recovering Oracle WebCenter Content . To recover from loss of host, see Recovering Oracle WebCenter Content to a Different Host.
Oracle WebCenter Content: Inbound Refinery	None	The Oracle home and the Administration Server domain directory. If the user data is not in the domain directory, back up the data.	The entity that has failed. If the user data needs to be recovered, recover it.	NA
Oracle WebCenter Content: Records	OCS schema	Depends on Oracle WebCenter Content and has no additional backup artifacts,	Depends on Oracle WebCenter Content and has no additional recovery artifacts,	NA
Oracle WebCenter Portal	MDS and WEBCENTER schemas	Administration Server domain	The Administration Server domain	NA
Oracle WebCenter Portal's analytics data	ACTIVITIES and MDS schema	The Oracle home and the domain home,	The Oracle home and the domain home.	NA
Oracle WebCenter Portal's discussion server	DISCUSSIONS schema	The Administration Server domain	The Administration Server domain	NA
Oracle WebCenter Portal's portlet producer	PORTLET schema	The Administration Server domain	The Administration Server domain	NA

Table 16-1 (Cont.) Backup and Recovery Recommendations

Component	Database Dependencies	Backup Recommendation	Recovery Recommendation	Additional Information
Oracle WebCenter Sites	WCSITES schema	The Oracle home and the Administration Server domain directory. If the WebCenter Sites configuration files are located outside of the Oracle home directory, back up those files as well. Ensure to back up WebCenter Sites shared filesystem (specified by the `wcsites.shared` property), as this contains all the files referenced by the database and should be backed up at the same time as the database backup.	The Oracle home and the Administration Server domain directory. If the WebCenter Sites configuration files are located outside of the Oracle home directory, restore those files as well. Ensure to restore WebCenter Sites shared filesystem (specified by the `wcsites.shared` property).	NA
Oracle WebLogic Server	By default, does not depend on any database repository. However, applications deployed on Oracle WebLogic Server may use databases as data sources.	The Oracle home and the Administration Server domain directory	The entity that has failed	If you use Whole Server Migration, see Recovering Oracle WebLogic Server with Whole Server Migration.
Oracle WebLogic Server JMS	Only if JMS is database-based	The Oracle home and the Administration Server domain directory	The entity that has failed	See Backup and Recovery Considerations for Oracle WebLogic Server JMS .

- Backup and Recovery Considerations for Oracle WebLogic Server JMS
- Backup and Recovery Recommendations for Oracle BPEL Process Manager

Backup and Recovery Considerations for Oracle WebLogic Server JMS

If you are using file-based JMS, use storage snapshot techniques for taking consistent online backups. Alternatively, you can use a file-system copy to perform an offline backup.

If the JMS persistent store is file-based, recover it from backup. If the JMS persistent store is database-based, recover the database to the most recent point in time, if needed. Note the following:

- Always try to keep JMS data as current as possible. This can be achieved by using the point-in-time recovery capabilities of Oracle Database, recovering to the most recent time (in the case of database-based persistence) or using a highly available RAID-backed storage device (for example, SAN/NAS).

- If you are using a file-based JMS, you can use storage snapshots to recover.
- If, for whatever reason, you need to restore JMS data to a previous point in time, there are potential implications. Restoring the system state to a previous point in time not only can cause duplicate messages, but can also cause lost messages. The lost messages are messages that were enqueued before or after the system restore point time, but never processed.

  Use the following procedure *before recovery* to drain messages in the JMS queue after persistent-store recovery to avoid processing duplicate messages:

  > **Note:**
  >
  > Do not drain and discard messages without first being certain that the messages contain no data that must be preserved. The recovered messages may include unprocessed messages with important application data, in addition to duplicate messages that have already been processed.

  1. Log into the Oracle WebLogic Remote Console.
  2. Before recovery, configure JMS server to pause Production, Insertion, and consumption operations at boot time to ensure that no new messages are produced or inserted into the destination or consumed from the destination before you drain stale messages. To do this:
     a. Expand **Services,** then **Messaging,** and then click **JMS Servers.**
     b. On the Summary of JMS Servers page, click the JMS server you want to configure for message pausing.
     c. On the Configuration: General page, click **Advanced** to define the message pausing options. Select **Insertion Paused At Startup, Production Paused At Startup,** and **Consumption Paused At Startup.**
     d. Click **Save.**

  Use the following procedure *after recovery*:

  1. After recovering the persistent store, start the Managed Servers.
  2. Drain the stale messages from JMS destinations, by taking the following steps:
     a. Expand **Services,** then **Messaging,** and then **JMS Modules.**
     b. Select a JMS module, then select a target.
     c. Select **Monitoring,** then **Show Messages.**
  3. Click **Delete All.**
  4. Resume operations, by taking the following steps:
     a. Expand **Services,** then **Messaging,** and then **JMS Servers.**
     b. On the Summary of JMS Servers page, click the JMS server you want to configure for message pausing.
     c. On the Configuration: General page, click **Advanced.** Deselect **Insertion Paused At Startup, Production Paused At Startup,** and **Consumption Paused At Startup.**
     d. Click **Save.**

If the store is not dedicated to JMS use, use the Oracle WebLogic Server JMS message management administrative tool. This tool can perform import, export, move, and delete operations from the Remote Console, MBeans, and WLST.

For applications that use publish and subscribe in addition to queuing, you should manipulate topic subscriptions in addition to queues.

Backup and Recovery Recommendations for Oracle BPEL Process Manager

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.

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 *Administering Oracle SOA Suite and Oracle Business Process Management Suite* .

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.

Assumptions and Restrictions for Backup and Recovery

Certain assumptions and restrictions apply to the backup and recovery procedures in this book.

Besides the following restrictions, also see the restrictions listed in Limitations and Restrictions for Backing Up Data.

- Only the user who installs the product or a user who has access privileges to the directories where Oracle Fusion Middleware has been installed should be able to execute backup and recovery operations.
- If a single Managed Server and Administration Server run on different hosts and the Managed Server is not in a cluster, you must use the pack and unpack commands on the Managed Server to retrieve the correct configuration.

  If you are using Cold Failover Cluster or Disaster Recovery, refer to Setting Up and Managing Disaster Recovery Sites in the *Disaster Recovery Guide* for additional information.

17

Backing Up Your Environment

Oracle provides recommended backup strategies for Oracle Fusion Middleware, with specific procedures for backing up your environment.

- Overview of Backup Strategies
  Backup strategies enable you safeguard your data and to later recover from critical failures that involve actual data loss.
- Limitations and Restrictions for Backing Up Data
  Certain limitations and restrictions apply when you are backing up data.
- Performing a Backup
  You can perform a full offline backup or an online or offline backup of run-time artifacts.
- 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.

Overview of Backup Strategies

Backup strategies enable you safeguard your data and to later recover from critical failures that involve actual data loss.

- Types of Backups
- Recommended Backup Strategy

Types of Backups

You can back up your Oracle Fusion Middleware environment offline or online:

- An **offline backup** means that you must shut down the environment before backing up the files. When you perform an offline backup, the Administration Server and all Managed Servers in the domain should be shut down.
- 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 Locking the WebLogic Server Configuration.

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.

Recommended Backup Strategy

The following 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. Perform a full offline backup at the following times:
  - Immediately after you install Oracle Fusion Middleware
  - Immediately before patching or upgrading your Oracle Fusion Middleware environment
  - Immediately before an operating system upgrade
  - Immediately after upgrading or patching Oracle Fusion Middleware

  See Performing a Full Offline Backup for information on performing a full backup.
- **Perform an online backup of run-time artifacts:** This involves backing up the run-time artifacts. 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:
  - After every administrative change and on a regular basis. Oracle recommends that you back up run-time artifacts nightly.
  - Prior to making configuration changes to a component.
  - After making configuration changes to a component.
  - Prior to deploying a custom Jakarta EE application to a Managed Server or cluster.
  - After a major change to the deployment architecture, such as creating servers or clusters.

  See Performing an Online Backup of Run-Time Artifacts for information on performing a backup of run-time artifacts.

  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 Locking the WebLogic Server Configuration.
- **Perform a new full backup after a major change**, such as any upgrade or patch, or if any of the following files are modified:

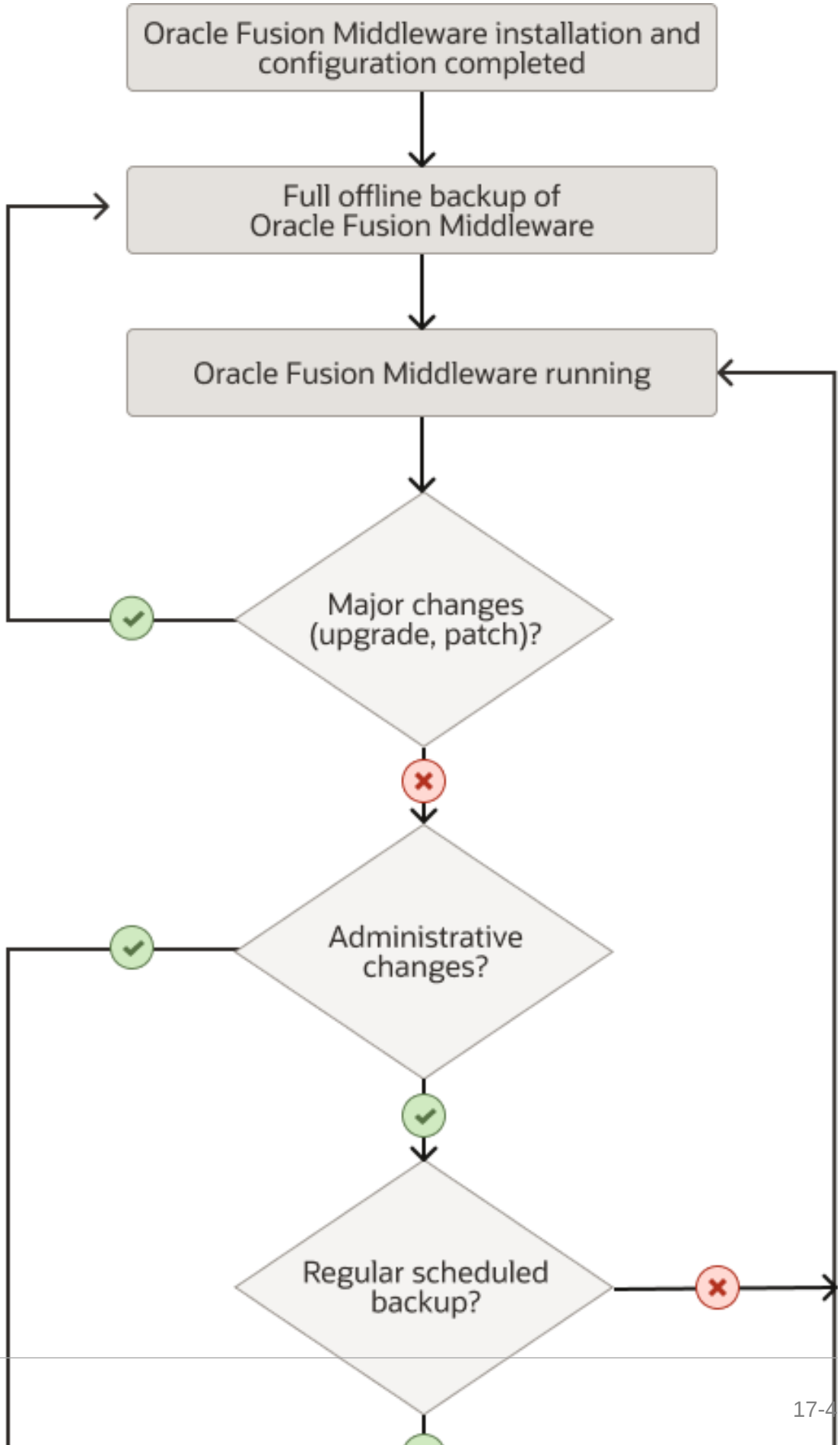
  ```
  DOMAIN_HOME/nodemanager/nodemanager.properties
  ORACLE_HOME/wlserver/common/bin/wlsifconfig.sh
  ORACLE_HOME/wlserver/common/bin/setPatchEnv.sh
  ORACLE_HOME/wlserver/common/bin/commEnv.sh
  ```

  See Performing a Full Offline Backup for information on performing a full backup.
- **Perform a full or incremental backup of your databases:** Use RMAN to backup your databases. See the *Oracle Database Backup and Recovery User's Guide* for information about using RMAN and for suggested methods of backing up the databases.
- Create a record of your Oracle Fusion Middleware environment. See Creating a Record of Your Oracle Fusion Middleware Configuration.
- When you create the backup, name the archive file with a unique name. Consider appending the date and time to the name. For example, if you create a backup of the Oracle home on June 5, 2017, name the backup:

```
oracle_home_backup_06052017.tar
```

The flowchart in Figure 17-1 provides an overview of how to decide which type of backup is appropriate for a given circumstance.

Figure 17-1 Decision Flow Chart for Type of Backup



Limitations and Restrictions for Backing Up Data

Certain limitations and restrictions apply when you are backing up data.

Note the following points:

- LDAP backups: If you use the built-in LDAP, do not update the configuration of a security provider while a backup of LDAP data is in progress. If a change is made (for example, if an administrator adds a user), while you are backing up the LDAP directory tree, the backups in the ldapfiles subdirectory could become inconsistent. Refer to *WebLogic Server Managing Server Startup and Shutdown* for detailed LDAP backup procedures.
- Java Transaction API (JTA): Oracle does not recommend that you back up and restore JTA transaction logs.
- Audit Framework: If you have configured Oracle Fusion Middleware Audit Framework to write data to a database, you should not back up the local files in the bus stop. (Auditable events from each component are stored in a repository known as a bus stop; each Oracle WebLogic Server has its own bus stop. Data can be persisted in this file, or uploaded to a central repository at which point the records are available for viewing and reporting.)

  If you back up the local files, duplicate records are uploaded to the database. That is, they are uploaded to the database when the bus stop is created and then are uploaded again when you restore the files.

  The default locations for bus stop local files are:

  – For Java components:

    DOMAIN_HOME/servers/*server_name*/logs/auditlogs/*component_type*

  – For system components, such as Oracle HTTP Server:

    DOMAIN_HOME/auditlogs/*component_type*/*component_name*

  For more information about Oracle Fusion Middleware Audit Framework and the bus stop, see Configuring and Managing Auditing in *Securing Applications with Oracle Platform Security Services*.
- Before you back up Oracle BI EE, you must lock the Oracle BI Presentation Catalogs so that the catalog and RPD remain synchronized. Run the following script:

  ORACLE_HOME/bi/bifoundation/OracleBIPresentationServicesComponent/coreapplication_obips1/catalogmanager/runcat.sh

  Use the following command:

  ```
  ./runcat.sh -cmd maintenanceMode -on -online OBIPS_URL
       -credentials credentials_properties_file
  ```

  After the backup is complete, turn off maintenance mode using the runcat command. For information on this command, see the help:

  ```
  ./runcat.sh -cmd maintenanceMode -help
  ```

Performing a Backup

You can perform a full offline backup or an online or offline backup of run-time artifacts.

- Performing a Full Offline Backup
- Performing an Online Backup of Run-Time Artifacts

- Backing Up Windows Registry Entries

Performing a Full Offline Backup

To perform a full offline backup, you copy the directories that contain Oracle Fusion Middleware files.

Archive and compress the source Oracle home, using your preferred tool for archiving, as described in Tools to Use for Backup and Recovery.

Take the following steps:

1. Shut down all processes in the Oracle home. For example, shut down the Managed Servers, the Administration Server, and any system components.
2. Back up the Oracle home (ORACLE_HOME) on all hosts. For example:

   ```
   (UNIX) tar -cf oracle_home_backup_06052017.tar ORACLE_HOME/*
   (Windows) jar cMf oracle_home_backup_06052017.jar ORACLE_HOME\*
   ```

3. Back up the Administration Server domain separately. This backs up Java components and any system components in the domain.

   For example:

   ```
   (UNIX) tar -cf domain_home_backup_06052017.tar DOMAIN_HOME/*
   (Windows) jar cMf domain_home_backup_06052017.jar DOMAIN_HOME\*
   ```

   In most cases, you do not need to back up the Managed Server directories separately, because the Administration Server domain contains information about the Managed Servers in its domain. If you have customized your environment for the Managed Server, back up the Managed Server directories. See Backup and Recovery Recommendations for Oracle Fusion Middleware Components for information about what you need to back up.
4. If a Managed Server is not located within the domain, back up the Managed Server directory. For example:

   ```
   (UNIX) tar -cf mg1_home_backup_06052017.tar server_name/*
   (Windows) jar cMf mgl_home_backup_06052017.jar server_name\*
   ```

5. Back up the application home directory. For example:

   ```
   (UNIX) tar -cf app_home_backup_06052017.tar Applications_Home/domain_name/*
   (Windows) jar cMf app_home_backup_06052017.jar Applications_Home\domain_name\*
   ```

6. Back up the OraInventory directory. For example:

   ```
   tar -cf Inven_home_backup_06052017.tar /scratch/oracle/OraInventory
   ```

7. On Linux and UNIX, back up the oraInst.loc file, which is located in the following directory:

   ```
   (Linux and IBM AIX) /etc
   (Other UNIX systems) /var/opt/oracle
   ```

8. On Linux and UNIX, backup the oratab file, which is located in the following directory:

   ```
   /etc
   ```

9. Back up the databases used in your environment using the Oracle Recovery Manager (RMAN). For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*.

   Make sure that the backup includes the SYSTEM.SCHEMA_VERSION_REGISTRY$ table.

Each Fusion Middleware schema has a row in SYSTEM.SCHEMA_VERSION_REGISTRY$ table. If you run the Upgrade Assistant to update an existing schema and it does not succeed, you must restore the original schema before you can try again. Make sure you back up your existing database schemas before you run the Upgrade Assistant.

10. On Windows, export the Windows Registry entries, as described in Backing Up Windows Registry Entries.
11. Unlock the WebLogic Server configuration.
12. Create a record of your Oracle Fusion Middleware environment. See Creating a Record of Your Oracle Fusion Middleware Configuration.

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 Recommended Backup Strategy.

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 Locking the WebLogic Server Configuration.
2. Back up the Administration Server domain directories. For example:

   ```
   UNIX) tar -cf domain_home_backup_06052017.tar DOMAIN_HOME/*
   (Windows) xcopy c:\DOMAIN_HOME e:\domain_home_backup_06052017 /s /e /i /h
   ```

   For Oracle Reports and Oracle Forms Services, you must back up the Managed Server directories, in addition to the Administration Server domain directories.
3. Back up the application home directory. For example:

   ```
   (UNIX) tar -cf app_home_backup_06052017.tar DOMAIN_HOME/*
   (Windows) jar cMf app_home_backup_06052017.jar C:\oracle\applications\domain_name\*
   ```
4. Back up the database repositories using the Oracle Recovery Manager (RMAN). For detailed steps, see the *Oracle Database Backup and Recovery User's Guide*.
5. Unlock the Oracle WebLogic Server configuration.
6. Create a record of your Oracle Fusion Middleware environment. See Creating a Record of Your Oracle Fusion Middleware Configuration.

Backing Up Windows Registry Entries

On Windows, you must back up Windows Registry keys related to Oracle Fusion Middleware. Which keys you back up depends on what components you have installed.

To export a key, use the following command:

```
regedit /E  FileName Key
```

Export the following entries:

- For any component, export the following registry key:

  ```
  HKEY_LOCAL_MACHINE\Software\Oracle
  ```

- For system components, such as Oracle HTTP Server, export each node that begins with **Oracle** within the following registry keys:

  ```
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
  HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services
  HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services
  ```

  For example:

  ```
  regedit /E C:\oracleSMP.reg HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services
  ```

  Use a unique file name for the each key.

- For Oracle BI EE, export the following registry key:

  ```
  HKEY_LOCAL_MACHINE\SOFTWARE\ODBC
  ```

  For example:

  ```
  regedit /E C:\oracleregistry.reg HKEY_LOCAL_MACHINE\SOFTWARE\ODBC
  ```

You can also use the Registry Editor to export the key, as described in the Registry Editor Help.

Creating a Record of Your Oracle Fusion Middleware Configuration

In the event that you need to restore and recover your Oracle Fusion Middleware environment, it is important to have all the necessary information at your disposal. This is especially true in the event of a hardware loss that requires you to reconstruct all or part of your Oracle Fusion Middleware environment on a new disk or host.

You should maintain an up-to-date record of your Oracle Fusion Middleware environment that includes the information listed in this section. You should keep this information both in hardcopy and electronic form. The electronic form should be stored on a host or e-mail system that is completely separate from your Oracle Fusion Middleware environment.

Your Oracle Fusion Middleware hardware and software configuration record should include:

- The following information for each host in your environment:
  - Host name
  - Virtual host name (if any)
  - Domain name
  - IP address
  - Hardware platform
  - Operating system release level and patch information
  - The version of the JDK and its path used in the installation and configuration of Oracle Fusion Middleware
- The following information for each Oracle Fusion Middleware installation in your environment:
  - Installation type (for example, Oracle HTTP Server)
  - Host on which the installation resides

  - User name, userid number, group name, groupid number, environment profile, and type of shell for the operating system user that owns the Oracle home (`/etc/passwd` and `/etc/group` entries)
  - Directory structure, mount points, and full path for the Oracle home, Oracle Common home, product homes, Oracle WebLogic Server domain home (if it does not reside in the user_projects directory in the Oracle home)
  - Amount of disk space used by the installation
  - Port numbers used by the installation
  - The version of the JDK and its path used in the installation and configuration of Oracle Fusion Middleware
- The following information for the database containing the metadata for components:
  - Host name
  - Database version and patch level
  - Base language
  - Character set
  - Global database name
  - SID
  - Listen port

18

Recovering Your Environment

Oracle provides recommended recovery strategies and procedures for recovering Oracle Fusion Middleware from different types of failures and outages, such as media failures or loss of host.

- Overview of Recovery Strategies
- Recovering After Data Loss, Corruption, Media Failure, or Application Malfunction
- Recovering After Loss of Host
  You need to recover your Oracle Fusion Middleware environment if you lose the original operating environment. For example, you could have a serious system malfunction or loss of media.

Overview of Recovery Strategies

Recovery strategies enable you to recover from critical failures that involve actual data loss.Depending on the type of loss, they can involve recovering any combination of the following types of files:

- Oracle software files
- Configuration files
- Oracle system files
- Windows Registry keys
- Application artifacts

You can recover your Oracle Fusion Middleware environment while Oracle Fusion Middleware is offline.

The following topics describe recovery strategies:

- Types of Recovery
- Recommended Recovery Strategies

Types of Recovery

You can recover your Oracle Fusion Middleware environment in part or in full. You can recover the following:

- The Oracle home
- WebLogic Server domains
- Standalone domains
- The Administration Server
- Managed Servers
- A component, such as Oracle SOA Suite or Oracle HTTP Server
- WebLogic Server cluster

- Deployed applications
- The database

Recommended Recovery Strategies

You should follow these recovery strategies for outages that involve actual data loss or corruption, host failure, or media failure where the host or disk cannot be restarted and they are permanently lost. This type of failure requires some type of data restoration before the Oracle Fusion Middleware environment can be restarted and continue with normal processing.

> **Note:**
>
> The procedures in this chapter assume that no administrative changes were made since the last backup. If administrative changes were made since the last backup, they must be reapplied after recovery is complete.

Note the following key points about recovery:

- Your Oracle Fusion Middleware environment must be offline while you are performing recovery.
- Rename important existing files and directories before you begin restoring the files from backup so that you do not unintentionally override necessary files.
- Although, in some cases, it may appear that only one or two files are lost or corrupted, you should restore the directory structure for the entire element, such as a domain, rather than just restoring one or two files. In this way, you are more likely to guarantee a successful recovery.
- Recover the database to the most current state, using point-in-time recovery (if the database is configured in Archive Log Mode). This is typically a time right before the database failure occurred.
- When you restore the files, use your preferred tool to extract the compressed files, as described in Tools to Use for Backup and Recovery.

  Ensure that the tool you are using preserves the permissions and timestamps of the files.

When you recover your environment, it is important to recover the entities in the correct order:

1. The database, if it needs to be recovered. See Recovering a Database and Recovering After Loss of Database Host.
2. The Oracle Home, if it needs to be recovered. See Recovering the Oracle Home.
3. The entire domain, if it needs to be recovered. See Recovering an Oracle WebLogic Server Domain and Recovering After Loss of Oracle WebLogic Server Domain Host for recovering a WebLogic Server managed domain. See Recovering a Standalone Domain for recovering a standalone domain.
4. The Administration Server, if you do not need to recover the domain. See Recovering the Administration Server Configuration and Recovering After Loss of Administration Server Host.
5. The Managed Servers, if they are not in the Administration Server domain directory and they need to be recovered. See Recovering a Managed Server and Recovering After Loss of Managed Server Host.

Java components are recovered when you recover the Managed Server. System components are recovered when you recover the domain. In some circumstances, you may need to take certain steps as described in Recovering a Component and Recovering After Loss of Component Host.

6. Some components require additional actions, which are described in the sections listed in Table 18-1.

Table 18-1 Additional Recovery Procedures for Particular Components

Component	For Data Loss, Corruption, Media Failure	For Loss of Host
Oracle B2B	*Recovering Oracle B2B*	*Recovering Oracle B2B*
Oracle Data Integrator	NA	Recovering Oracle Data Integrator to a Different Host
Oracle Forms Services	NA	No additional steps needed if recovering to the same host. To recover to a different host, see Recovering Oracle Forms Services to a Different Host.
Oracle HTTP Server	NA	Recovering Web Tier Components to a Different Host
Oracle Platform Security Services	Recovering Oracle Platform Security Services	Recovering Oracle Platform Security Services
Oracle Reports	NA	Recovering Oracle Reports to a Different Host
Oracle SOA Suite	NA	No additional steps needed if recovering to the same host. To recover to a different host, see Recovering Oracle SOA Suite After Loss of Host.
Oracle WebCenter Content	Recovering Oracle WebCenter Content	Recovering Oracle WebCenter Content to a Different Host
Oracle WebCenter Content: Records	Recover Oracle WebCenter Content. See Recovering Oracle WebCenter Content .	Recover Oracle WebCenter Content. See Recovering Oracle WebCenter Content to a Different Host.
Oracle WebCenter Portal Analytics	Recovering Oracle WebCenter Portal's Analytics	Recovering Oracle WebCenter Portal's Analytics
Oracle WebLogic Server	For Oracle WebLogic Server with whole server migration, see Recovering Oracle WebLogic Server with Whole Server Migration.	For Oracle WebLogic Server with whole server migration, see Recovering Oracle WebLogic Server with Whole Server Migration.

7. Applications, if they need to be recovered. See Recovering Applications.

Recovering After Data Loss, Corruption, Media Failure, or Application Malfunction

You need to recover some or all of your environment in cases of outages that involve actual data loss or corruption or media failure where the disk cannot be restored. You may also need to recover 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.

> **Note:**
>
> You can only restore an entity to the same path as the original entity. That path can be on the same host or a different host.

- Recovering the Oracle Home
- Recovering an Oracle WebLogic Server Domain
- Recovering a Standalone Domain
- Recovering the Administration Server Configuration
- Recovering a Managed Server
- Recovering a Component
- Recovering a Cluster
- Recovering Applications
- Recovering a Database

Recovering the Oracle Home

You can recover the Oracle home that was corrupted or from which files were deleted.

To recover the Oracle home:

1. Stop all relevant processes. That is, stop all processes that are related to the domain, such as the Administration Server, Node Manager, and Managed Servers. For example, to stop the Administration Server on Linux:

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

2. Change to the directory that you want to be the parent directory of the Oracle home directory. Use the same directory structure as in the original environment.
3. Recover the Oracle home directory from backup. For example:

   ```
   (UNIX) tar -xf oracle_home_backup_06052017.tar
   (Windows) jar xf oracle_home_backup_06052017.jar
   ```

4. Start all relevant processes. That is, start all processes that run in the Oracle home, such as the Administration Server and Managed Servers. For example, start the Administration Server:

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

Recovering an Oracle WebLogic Server Domain

You can recover an Oracle WebLogic Server domain that was corrupted or deleted from the file system, or when the host containing the domain was lost.

> **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. If any relevant processes are running, stop them. That is, stop all processes that are related to the domain, such as the Administration Server, Managed Servers, and any system components. For example, stop the Administration Server:

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

   For information on stopping system components such as Oracle HTTP Server, see Starting and Stopping Components Using the Command Line.
2. Change to the directory that you want to be the parent directory of the domain home directory. Use the same directory structure as in the original environment.
3. Recover the domain directory from backup:

   ```
   (UNIX) tar -xf domain_backup_06052017.tar
   (Windows) jar xf domain_backup_06052017.jar
   ```
4. Start all relevant processes. That is, start all processes that are related to the domain, such as the Administration Server, Managed Servers, and any system components. For example, start the Administration Server:

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

   For information on starting system components such as Oracle HTTP Server, see Starting and Stopping Components Using the Command Line.
5. If you cannot start the Administration Server, recover it, as described in Recovering the Administration Server Configuration.
6. If you cannot start a Managed Server, recover it, as described in Recovering a Managed Server.

- Recovering Oracle WebLogic Server with Whole Server Migration

Recovering Oracle WebLogic Server with Whole Server Migration

When using database leasing (for example, with whole server migration), if you recover Oracle WebLogic Server, you should discard the information in the leasing table. You can simply drop and recreate the leasing table by running the leasing table creation script. (For more information about Whole Server Migration, see Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.)

Recovering a Standalone Domain

You can recover a standalone domain that contains system components, such as Oracle HTTP Server, that was corrupted or deleted from the file system or if the host was lost and you want to recover to the same host.

To recover a standalone domain:

1. If Node Manager or a system component, such as Oracle HTTP Server are running, stop them.
2. If it is corrupted, recover the Oracle home:

```
(UNIX) tar xf oracle_home_backup_05_21_2013.tar
(Windows) jar xf oracle_home_backup_05_21_2013.jar
```

3. Recover the domain home:

```
(UNIX) tar xf domain_backup_05_21_2013.tar
(Windows) jar xf domain_backup_05_21_2013.jar
```

4. Start Node Manager:

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

5. Start any system components, such as Oracle HTTP Server, that are in the domain:

```
(UNIX) Domain_Home/bin/startComponent.sh ohs1
(Windows) Domain_Home\bin\startComponent.cmd ohs1
```

Recovering the Administration Server Configuration

If the Administration Server configuration has been lost because of file deletion or file system corruption, the Remote Console continues to function if it was already started when the problem occurred. To prevent the Administration Server from prompting for a user name and password, see Enabling Servers to Start Without Supplying Credentials.

> **Caution:**
>
> Performing a domain-level recovery can impact other aspects of a running system and all of the configuration changes performed after the backup was taken will be lost.

To recover the Administration Server configuration:

1. Stop all processes, including the Administration Server, Managed Servers, and Node Manager, if they are started. For example, to stop the Administration Server:

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

2. Recover the Administration Server configuration by recovering the domain home backup to a temporary location. Then, restore the config directory to the following location:

```
DOMAIN_HOME/config
```

3. Start the Administration Server. For example:

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

4. Verify that the Administration Server starts properly and is accessible.

On the next configuration change, the configuration from the Administration Server is pushed to the Managed Servers. On each Managed Server restart, the configuration is retrieved from the Administration Server.

Recovering a Managed Server

You can recover a Managed Server's files, including its configuration files if they are deleted or corrupted.

In this scenario, the Managed Server is not on the same host as the Administration Server, and it does not operate properly or cannot be started because the configuration has been deleted or corrupted or the configuration was mistakenly changed and you cannot ascertain what was changed.

To recover a Managed Server:

1. If the Administration Server is not reachable, recover the Administration Server, as described in Recovering the Administration Server Configuration.
2. If the Managed Server is running, stop it. For example:

   ```
   DOMAIN_HOME/bin/stopManagedWeblogic.sh managed_server_name admin_url
           username password
   ```

3. Recover the Oracle home from the backup, if required. For example:

   ```
   tar -xf oracle_home_backup_06052017.tar
   ```

4. Stop Node Manager as described in Starting and Stopping Node Manager.
5. Create a domain template jar file for the Administration Server, using the pack utility on the Administration Server host. For example:

   ```
   pack.sh -domain=/scratch/oracle/config/domains/WLS_domain
     -template=/scratch/temp.jar -template_name=test_install
     -template_author=myname -log=/scratch/logs/my.log -managed=true
   ```

   Specifying the -managed=true option packs up only the Managed Servers. If you want to pack the entire domain, omit this option.
6. Unpack the domain template jar file, using the unpack utility on the Managed Server host. In the following example, temp.jar is the archive created by the pack command:

   ```
   unpack.sh -template=/scratch/temp.jar
      -domain=/scratch/oracle/config/domains/WLS_domain
      -log=/scratch/logs/new.log -log_priority=info
   ```

> **Note:**
>
> - The following directory must exist. If it does not, the unpack command fails.
>
>   ```
>   ORACLE_HOME/config/domains/
>   ```
>
> - The unpack command provides an -overwrite_domain option, which allows unpacking a Managed Server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. Use the -overwrite_domain option, if required for your deployment.
> - By default, applications are stored in the following directory unless you pass another location using the -app_dir argument:
>
>   ```
>   ORACLE_HOME/user_projects/applications/Domain_Name
>   ```

7. Start the Managed Server. For example:

   ```
   DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
   ```

   The Managed Server connects to the Administration Server and updates its configuration changes.

Recovering a Component

You can recover a component if the component's files have been deleted or corrupted or if the component cannot be started or is not functioning properly because the component's configuration was changed and committed. You may not be able to ascertain what change is causing the problem and you want to revert to an earlier version.

- For Java components, you recover the Managed Server, as described in Recovering a Managed Server.
- For system components, such as Oracle HTTP Server, in a standalone domain, you recover the domain, as described in Recovering a Standalone Domain.
- For system components, such as Oracle HTTP Server, in an Oracle WebLogic Server domain, you recover the domain as described in Recovering an Oracle WebLogic Server Domain.

The following topics describes additional steps you must take for certain components:

- Recovering Oracle Platform Security Services
- Recovering Oracle WebCenter Portal's Analytics
- Recovering Oracle WebCenter Content

Recovering Oracle Platform Security Services

For Oracle Platform Security Services, restore the following files:

```
DOMAIN_HOME/config/fmwconfig/jps-config.xml
DOMAIN_HOME/config/fmwconfig/jps-config-jse.xml
DOMAIN_HOME/config/fmwconfig/cwallet.sso
DOMAIN_HOME/config/fmwconfig/bootstrap/cwallet.sso
DOMAIN_HOME/config/fmwconfig/keystores.xml
DOMAIN_HOME/config/config.xml
DOMAIN_HOME/config/fmwconfig/ids_config.xml
DOMAIN_HOME/config/fmwconfig/system-jazn-data.xml (if present)
DOMAIN_HOME/config/fmwconfig/jps_mbeans.xml
```

Recovering Oracle WebCenter Portal's Analytics

To recover Oracle WebCenter Portal's Analytics:

1. Restore the domain, as described in Recovering an Oracle WebLogic Server Domain.
2. Restore the Oracle home, as described in Recovering the Oracle Home.
3. Restore the database containing the ACTIVITIES and MDS schemas, if necessary.

Recovering Oracle WebCenter Content

To recover Oracle WebCenter Content:

1. If necessary, restore the database, as described in Recovering After Loss of Database Host.

2. Restore the domain, as described in Recovering an Oracle WebLogic Server Domain.
3. If the Vault, WebLayout, or Search directories are not located in the domain directory, restore those directories, if necessary. For example, if the Vault directory is located on a shared drive in /net/home/vault, restore it from backup:

```
cd /net/home/vault
tar -xf vault_backup_042012.tar
```

Note that you should restore the database and the shared file system at the same time. If you cannot do that, you can use the IDCAnalyse utility to determine if there are any inconsistencies between the database and the shared file system. If there are, you can perform a manual recovery using IDCAnalyse.

Recovering a Cluster

You may need to recover a cluster in the following situations:

- The cluster has been erroneously deleted or a cluster member was erroneously deleted.
- The cluster-level configuration, such as the JMS configuration or container-level data sources, was mistakenly changed and committed. The component or server cannot be started or does not operate properly or the services running inside the server are not starting. 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.

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 Fusion Middleware Control, the Oracle WebLogic Remote Console, or WLST. For example, to use WLST:

```
shutdown('cluster_name', 'Cluster')
```

2. Stop all processes, such as the Administration Server and Managed Servers. For example, to stop the Administration Server:

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

3. Recover the Administration Server configuration by recovering the domain home backup to a temporary location. Then, restore the config directory to the following location:

```
DOMAIN_HOME/config
```

4. Start the Administration Server. For example:

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

   Any deleted members are now back in the cluster.

5. Start all processes, such as the Managed Servers. For example, to start the Managed Server on Linux, use the following script:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

6. Start the cluster. You can use Fusion Middleware Control, the Oracle WebLogic Remote Console, or WLST. For example, to use WLST:

   ```
   start('cluster_name', 'Cluster')
   ```

Recovering Applications

Note the following about recovering applications:

- If the application is staged, the Administration server copies the application bits to the staged directories on the Managed Server hosts.
- If the deployment mode is nostage or external_stage, ensure that additional application artifacts are available. For example, applications may reside in directories outside of the domain directory. Make your application files available to the new Administration Server by copying them from backups or by using a shared disk. Your application files should be available in the same relative location on the new file system as on the file system of the original Administration Server.

  See *Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

The following topics describe how to recover an application:

- Recovering Application Artifacts
- Recovering a Jakarta EE Application

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:

- Start the Managed Server to which the application was deployed. For example:

  ```
  DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
  ```

  This synchronizes the configuration with the Administration Server.

  On each Managed Server restart, the configuration and application artifacts are retrieved from the Administration Server.

Recovering a Jakarta EE Application

You can recover a Jakarta EE application:

- If a Jakarta 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.
- If a deployed application was undeployed from Oracle WebLogic Server.
- A new version of a composite application was redeployed to a Managed Server or cluster. The application is no longer functional.

To recover the application:

1. Recover the application files from backup, if needed.
2. Redeploy the old version of the application 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.

Recovering a Database

If your database that contains your metadata repository, including the MDS Repository, is corrupted, you can recover it using RMAN. You can recover the database at the desired granularity, either a full recovery or a tablespace recovery.

For best results, recover the database to the most current state, using point-in-time recovery (if the database is configured in Archive Log Mode.) This ensures that the latest data is recovered. For example:

```
rman> restore database;
rman> recover database;
```

See *Creating Schemas with the Repository Creation Utility* for the schemas used by each component.

For detailed steps for recovering a database, see the *Oracle Database Backup and Recovery User's Guide*.

Recovering After Loss of Host

You need to recover your Oracle Fusion Middleware environment if you lose the original operating environment. For example, you could have a serious system malfunction or loss of media.

- Recovering After Loss of Oracle WebLogic Server Domain Host
- Recovering After Loss of Standalone Domain Host
- Recovering After Loss of Administration Server Host
- Recovering After Loss of Managed Server Host
- Recovering After Loss of Component Host
- Additional Actions for Recovering Entities After Loss of Host
- Recovering After Loss of Database Host

Recovering After Loss of Oracle WebLogic Server Domain Host

To recover an Oracle WebLogic Server domain after loss of host, follow the steps in Recovering an Oracle WebLogic Server Domain.

Recovering After Loss of Standalone Domain Host

If you lose a host that contains a standalone domain, you can recover it to the same host or a different host, as described in the following topics:

- Recovering a Standalone Domain to the Same Host
- Recovering a Standalone Domain to a Different Host

Recovering a Standalone Domain to the Same Host

To recover the standalone domain to the same host after the operating system has been reinstalled, follow the procedures in Recovering a Standalone Domain.

Recovering a Standalone Domain to a Different Host

In this scenario, you recover the standalone domain to a different host.

To recover the standalone domain to a different host:

1. Recover the Oracle home:

```
(UNIX) tar xf oracle_home_backup_05_21_2017.tar
(Windows) jar xf oracle_home_backup_05_21_2017.jar
```

2. Recover the domain home:

```
(UNIX) tar xf domain_backup_05_21_2017.tar
(Windows) jar xf domain_backup_05_21_2017.jar
```

3. In a standalone domain, by default, Node Manager is listening on localhost. However, if it is not, you can update the ListenAddress by using the following WLST commands:

```
readDomain('Domain_Home')
cd('/')
cd('NMProperties')
set('ListenAddress','localhost')
set('ListenPort',9001)
updateDomain()
```

4. Start Node Manager:

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

5. Update any system component configuration files manually.

   See Recovering After Loss of Component Host for details for specific components.

6. Start any system components, such as Oracle HTTP Server, that are in the domain. For example:

```
(UNIX) Domain_Home/bin/startComponent.sh ohs1
(Windows) Domain_Home\bin\startComponent.cmd ohs1
```

7. Update the Oracle Inventory, as described in Updating Oracle Inventory.
8. For Windows, update the Windows Registry, as described in Recovering the Windows Registry.

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

Recovering the Administration Server to the Same Host

In this scenario, you recover the Administration Server either to the same host after the operating system has been reinstalled or to a new host that has the same host name. For example, the Administration Server is running on Host A and the Managed Server is running on Host B. Host A has failed for some reason and the Administration Server must be recovered.

To recover the Administration Server to the same host:

1. Recover the file system. For example, recover the domain containing the Administration Server, as described in Recovering After Loss of Oracle WebLogic Server Domain Host.
2. Start the Administration Server. For example:

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

   If the Administration Server starts, you do not need to take any further steps.
3. If the Administration Server fails to start, take the following steps on Host A:
   a. Stop all relevant processes. That is, stop all processes that are related to the domain, such as the Managed Servers.
   b. Recover the Oracle home, if needed:

      ```
      tar -xf oracle_home_backup_06052017.tar
      ```

   c. If the domain directory does not reside in the Oracle home, recover the domain directory from backup. First, change to the directory that you want to be the parent of the Domain home, then:

      ```
      tar -xf domain_backup_06052017.tar
      ```

   d. Start the Administration Server. For example:

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

   e. Start the Managed Servers, specifying the Administration URL for the host:

      ```
      DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
      ```

   f. Start Node Manager:

      ```
      DOMAIN_HOME/bin/startNodeManager.sh
      ```

Recovering the Administration Server to a Different Host

In this scenario, the Administration Server is running on Host A and the Managed Server is running on Host B. Host A has failed for some reason and the Administration Server must be moved to Host C.

To recover the Administration Server to a different host:

1. Recover the Oracle home to Host C (the new Host):

   ```
   tar -xf oracle_home_backup_06052017.tar
   ```

2. If the domain directory does not reside in the Oracle home, recover the domain directory from backup. First, change to the directory that you want to be the parent of the Domain home, then:

   ```
   tar -xf domain_backup_06052017.tar
   ```

3. If the Administration Server has a Listen address, create a new machine with the new host name, as described in Creating a New Machine for Certain Components.
4. Start the Administration Server. For example:

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

5. Using WLST, connect to the Administration Server and then enroll Node Manager running in the new host with the Administration Server:

   ```
   connect('username','password','t3://host:port')
   nmEnroll('/scratch/oracle/config/domains/domain_name',
     'DOMAIN_HOME/nodemanager')
   ```

   Note that on Windows, as on UNIX, you use slashes (/), not backslashes (\), in the nmEnroll command.

6. Edit the Node Manager properties file, changing the Listen Address property. For a domain-based Node Manager, the file is located at:

   ```
   DOMAIN_HOME/nodemanager/nodemanager.properties
   ```

   Alternatively, you can use the following WLST commands to change the property:

   ```
   readDomain('Domain_Home')
   cd('/')
   cd('NMProperties')
   set('ListenAddress','localhost')
   set('ListenPort',port_num)
   updateDomain()
   ```

7. Start Node Manager on Host C if it was configured on the original host:

   ```
   cd DOMAIN_HOME/bin
   ./startNodeManager.sh
   ```

8. Start the Managed Servers. The section Restarting a Failed Administration Server in *Administering Server Startup and Shutdown for Oracle WebLogic Server* describes different ways to restart them, depending on how they were configured.

   One option is to use the following script, specifying the Administration URL of the new host:

   ```
   DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
   ```

9. Ensure that additional application artifacts are available. For example, if the deployment mode is nostage or external_stage, applications may reside in directories outside of the domain directory. Make your application files available to the new Administration Server by copying them from backups or by using a shared disk. Your application files should be available in the same relative location on the new file system as on the file system of the original Administration Server.

   If the application is staged, the Administration Server copies the application bits to the staged directories on the Managed Server hosts.

10. Update Oracle Inventory, as described in Updating Oracle Inventory.
11. On Windows, recover the Windows Registry, as described in Recovering the Windows Registry
12. If your environment contains Oracle HTTP Server, modify the mod_wl_ohs.conf file, as described in Modifying the mod_wl_ohs.conf File.

Now you can start and stop the Managed Server on Host B using the Remote Console running on Host C.

If you are recovering the Administration Server for a Web Tier installation, see Additional Actions for Recovering Entities After Loss of Host for information about additional actions you must take.

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 a Managed Server to the Same Host

In this scenario, you recover a Managed Server to the same host after the operating system has been reinstalled or to a new host that has the same host name. The Administration Server is running on Host A and the Managed Server is running on Host B. Host B failed for some reason and the Managed Server must be recovered to Host B.

To recover a Managed Server to the same host:

1. Start Node Manager on Host B:

   ```
   cd DOMAIN_HOME/bin
   ./startNodeManager.sh
   ```

2. Start the Managed Server. For example:

   ```
   DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
   ```

   If the Managed Server starts, it connects to the Administration Server and updates its configuration changes. You do not need to take any further steps.

3. If the Managed Server fails to start or if the file system is lost, take the following steps:

   a. Recover the Oracle home to Host B from the backup, if required:

      ```
      tar -xf oracle_home_backup_06052017.tar
      ```

   b. Stop Node Manager as described in Starting and Stopping Node Manager.

   c. If the Managed Server contains Oracle Reports or Oracle Forms Services, and the Managed Server domain directories reside outside of the Oracle home, restore the domain, in addition to the Oracle home. For example:

      ```
      cd Domain_Home
      tar -xf domain_home_backup_06052017.tar
      ```

      Go to Step 3.e.

   d. If the Managed Server does not contain Oracle Forms Services or Oracle Reports, 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=/scratch/oracle/config/domains/domain_name
          -template=/scratch/temp.jar -template_name=test_install
          -template_author=myname -log=/scratch/logs/my.log -managed=true
        ```

        Specifying the -managed=true option packs up only the Managed Servers. If you want to pack the entire domain, omit this option.

- Unpack the domain template jar file in Host B, using the unpack utility:

```
unpack.sh -template=/scratch/temp.jar
  -domain=/scratch/oracle/config/domains/domain_name
  -log=/scratch/logs/new.log -log_priority=info
```

e. Ensure that the application artifacts are accessible from the Managed Server host. That is, if the application artifacts are not on the same server as the Managed Server, they must be in a location accessible by the Managed Server.

> **Note:**
>
> - For applications that are deployed in nostage and external_stage mode, copy the application artifacts from the Administration Server host directory.
> - For applications that are deployed in stage mode, the Administration server copies the application bits to the staged directories on the Managed Server hosts.
>
> See *Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

f. Update the Node Manager property ListenAddress by using the following WLST commands:

```
readDomain('Domain_Home')
cd('/')
cd('NMProperties')
set('ListenAddress','localhost')
set('ListenPort',9001)
updateDomain()
```

g. If Node Manager is not started, start it:

```
cd DOMAIN_HOME/bin
./startNodeManager.sh
```

h. Start the Managed Server. For example:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

The Managed Server connects to the Administration Server and updates its configuration changes.

Recovering a Managed Server to a Different Host

In this scenario, the Administration Server is running on Host A and the Managed Server is running on Host B. Host B failed for some reason and the Managed Server must be recovered to Host C. There are two machines, which are logical representations of the computer that hosts one or more WebLogic Servers, machine_1 on Host A and machine_2 on Host B.

> **Note:**
>
> Recover the Oracle home to the same location as the original.

To recover a Managed Server to a different host:

1. Recover the Oracle home for the Managed Server to Host C.

   ```
   tar -xf oracle_home_backup_06052014.tar
   ```

2. Reconfigure the topology to point to the new host:

   a. To avoid an inconsistent backup, do not make any configuration changes until the backup is completed. To ensure that no changes are made in the WebLogic Server domain, lock the WebLogic Server configuration, as described in Locking the WebLogic Server Configuration.

   b. In Fusion Middleware Control, change the configuration of machine_2, to point it to the new host:

      From the WebLogic Domain menu, expand **Environment** and then select **Machines**. On the Machine page, select machine_2 and select the Configuration tab. Then select the Node Manager tab. Change the **Listen Address** to the address for Host C. Click **Save.**

      If you identify the Listen Address by IP address, you must disable Host Name Verification on the Administration Servers that access Node Manager. For more information and instructions, see Using Hostname Verification in *Administering Security for Oracle WebLogic Server*.

   c. Change the Managed Server configuration to point to the new host:

      From the left pane of the Remote Console, expand **Environment** and then select **Servers**. The, select the name of the server. Select the **Configuration** tab, then the **General** tab.

      Change the **Machine** to machine_2.

      Change **Listen Address** to the new host. (If the listening address was set to blank, you do not need to change it.)

      Click **Save,** then click **Activate Changes.**

   d. Unlock the Oracle WebLogic Server configuration by clicking **Release Configuration** on the Edit Session menu.

3. Take any additional steps needed for components as described in Table 18-1.

4. Stop Node Manager as described in Starting and Stopping Node Manager.

5. If the Managed Server contains Oracle Reports or Oracle Forms Services, and the Managed Server domain directories reside outside of the Oracle home, restore the domain, in addition to the Oracle home. For example:

   ```
   cd Domain_Home
   tar -xf domain_home_backup_042012.tar
   ```

   Go to Step 7.

6. If the Managed Server does not contain the components listed in Step 5, 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=/scratch/oracle/config/domains/domain_name
        -template=/scratch/temp.jar -template_name=test_install
        -template_author=myname -log=/scratch/logs/my.log -managed=true
      ```

Specifying the -managed=true option packs up only the Managed Servers. If you want to pack the entire domain, omit this option.

b. Unpack the domain template jar file on Host C, using the unpack utility:

```
unpack.sh -template=/scratch/temp.jar
   -domain=/scratch/oracle/config/domains/domain_name
   -log=/scratch/logs/new.log -log_priority=info
```

If you are recovering to a different domain home, use the -app_dir switch in the unpack command.

7. Ensure that the application artifacts are accessible from the Managed Server host. That is, if the application artifacts are not on the same server as the Managed Server, they must be in a location accessible by the Managed Server.

> **Note:**
>
> - For applications that are deployed in nostage and external_stage mode, copy the application artifacts from the Administration Server host directory.
> - For applications that are deployed in stage mode, the Administration server copies the application bits to the staged directories on the Managed Server hosts.
>
> See *Deploying Applications to Oracle WebLogic Server* for information about deploying applications.

8. Update the ListenAddress by using the following WLST commands:

```
readDomain('Domain_Home')
cd('/')
cd('NMProperties')
set('ListenAddress','localhost')
set('ListenPort',9001)
updateDomain()
```

9. Start Node Manager on Host C, if it is not started:

```
cd DOMAIN_HOME/bin
./startNodeManager.sh
```

10. Start the Managed Server. For example:

```
DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

The Managed Server connects to the Administration Server and updates its configuration changes.

11. Update Oracle Inventory, as described in Updating Oracle Inventory.
12. On Windows, recover the Windows Registry, as described in Recovering the Windows Registry
13. If your environment contains Oracle HTTP Server, modify the mod_wl_ohs.conf file, as described in Modifying the mod_wl_ohs.conf File.

Now you can start and stop the Managed Server on Host C using the Administration Server running on Host A.

Recovering After Loss of Component Host

If you lose a host that contains a component (and its Managed Server, if applicable), you can recover most components to the same host or a different host using the procedures described in the following topics:

Some components require additional actions, which are described in the sections listed in Table 18-1.

- Recovering a Java Component to the Same or Different Host
- Recovering a Java Component to a Different Host
- Recovering a System Component to the Same or Different Host
- Recovering Oracle SOA Suite After Loss of Host
- Recovering Web Tier Components to a Different Host
- Recovering Oracle Forms Services to a Different Host
- Recovering Oracle Reports to a Different Host
- Recovering Oracle Data Integrator to a Different Host
- Recovering Oracle WebCenter Content to a Different Host

Recovering a Java Component to the Same or Different Host

To recover a Java component to the same host:

1. Recover the Managed Server, as described in Recovering a Managed Server to the Same Host.
2. If the component requires additional steps, as noted in Table 18-1, take those steps.

Recovering a Java Component to a Different Host

To recover a Java component to a different host:

1. Recover the Managed Server, as described in Recovering a Managed Server to a Different Host.
2. If the component requires additional steps, as noted in Table 18-1, take those steps.

Recovering a System Component to the Same or Different Host

To recover a system component, such as Oracle HTTP Server, to the same host or a different host:

- For system components, such as Oracle HTTP Server, in a standalone domain, you recover the domain, as described in Recovering After Loss of Standalone Domain Host.
- For system components, such as Oracle HTTP Server, in an Oracle WebLogic Server domain, you recover the domain, as described in Recovering After Loss of Oracle WebLogic Server Domain Host.

However, some components require additional steps, as noted in Table 18-1.

Recovering Oracle SOA Suite After Loss of Host

To recover the Oracle SOA Suite Managed Server to the same host, recover the Managed Server, as described in Recovering a Managed Server to the Same Host.

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 Recovering a Managed Server to a Different Host.
3. After you recover the Oracle SOA Suite Managed Server, take the following actions:
   - 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 must 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.

4. If a Load Balancer is configured with the environment, take the following additional steps:
   a. In Fusion Middleware Control, from the WebLogic Domain menu, select **Environment,** then **Clusters.**
   b. Select the cluster you want to configure.
   c. From the WebLogic Cluster menu, select **Administration,** then **HTTP.**
   d. For **Frontend Host,** enter the new host name.
   e. For **Frontend HTTP Port** and **Frontend HTTPs Port,** if applicable, enter the new port number.
   f. Restart each Managed Server.

Recovering Web Tier Components to a Different Host

The Web tier consists of Oracle HTTP Server. The following topics describe how to recover it to a different host:

- Recovering Oracle HTTP Server in a Standalone Domain to a Different Host
- Recovering Oracle HTTP Server in a WebLogic Server Domain to a Different Host

Recovering Oracle HTTP Server in a Standalone Domain to a Different Host

To recover Oracle HTTP Server in a standalone domain:

1. Follow steps 1 through 4 in Recovering After Loss of Standalone Domain Host.
2. Update the configuration files in the following directory:

   ```
   (UNIX) DOMAIN_HOME/config/fmwconfig/components/OHS/instance_name
   (Windows) DOMAIN_HOME\config\fmwconfig\components\OHS\instance_name
   ```

   For example, update the IP Address and host name in httpd.conf, admin.conf, and mod_wl_ohs.conf (if required).
3. Follow steps 6 through 8 in Recovering After Loss of Standalone Domain Host.

Recovering Oracle HTTP Server in a WebLogic Server Domain to a Different Host

To recover Oracle HTTP Server in an Oracle WebLogic Server domain to a different host:

1. Recover the domain, as described in Recovering After Loss of Oracle WebLogic Server Domain Host.
2. Change the configuration of the Oracle HTTP Server instance that was on Host B:
   a. In Fusion Middleware Control, from the navigation pane, expand **HTTPServer.**
   b. Select the Oracle HTTP Server instance, such as ohs1.
   c. From the Oracle HTTP Server menu, select **Administration,** then **Ports Configuration.**
   d. For each port in the table, select the port, then click **Edit.** Change the **IP Address.**

      Note that if ANY is selected, you do not need to make any changes.
   e. Click **OK.**
3. Update the mod_wl_ohs wiring for each Oracle HTTP Server instance:
   a. In Fusion Middleware Control, from the navigation pane, expand **HTTP Server.**
   b. Select the Oracle HTTP Server instance, such as ohs1.
   c. From the Oracle HTTP Server menu, select **Administration,** then **mod_wl_ohs Configuration.**
   d. In the Locations section, click **AutoFill.**

      All valid WebLogic Server endpoint locations are displayed.
   e. Click **Apply.**
4. Restart any Oracle HTTP Server instances that are not on the failed machine by navigating to that instance and clicking **Start Up.**
5. Start the Oracle HTTP Server instances on Host C by navigating to that instance and clicking **Start Up.**

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 Recovering a Managed Server to a Different Host.

2. 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
   ```

3. Copy the file from the previous step to the new host.

4. In the new host, modify the OssoConfigFile section in the following file to include the path of the file in step 2:

   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
   ```

Recovering Oracle Reports to a Different Host

To recover Oracle Reports to a different host:

1. Recover the Managed Server, as described in Recovering a Managed Server to a Different Host.

2. Edit the following files, replacing the previous host name with the new host name:

   - nodemanager.properties. The file is located in:

     ```
     (UNIX) DOMAIN_HOME/nodemanager
     (Windows) DOMAIN_HOME\nodemanager
     ```

   - reports_ohs.conf. The file is located in:

     ```
     (UNIX) DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs_name/moduleconf
     (Windows)
     DOMAIN_HOME\config\fmwconfig\components\OHS\instances\ohs_name\moduleconf
     ```

   - rwservlet.properties. The file is located in:

     ```
     (UNIX) DOMAIN_HOME/config/fmwconfig/servers/server_name/applications/
     reports_version/configuration
     (Windows)
     DOMAIN_HOME\config\fmwconfig\servers\server_name\applications\reports_version\con
     figuration
     ```

     In the file, modify the <server> element to use the new host name.

   - mbeans.xml. The file is located in:

```
(Unix) DOMAIN_HOME/servers/server_name/tmp/_WL_user/reports_version/
random_string/META-INF
(Windows)
DOMAIN_HOME\servers\server_name\tmp\_WL_user\reports_version\random_string\META-
INF
```

Recovering Oracle Data Integrator to a Different Host

To recover Oracle Data Integrator, follow the procedures in one or both of these topics, depending on the failure:

- Recovering Oracle Data Integrator Repository
- Recovering Oracle Data Integrator Agents to a Different Host

Recovering Oracle Data Integrator Repository

If the Oracle Data Integrator Repository must be restored to a different host:

1. Restore the database, as described in Recovering After Loss of Database Host.
2. Connect to the restored Oracle Data Integrator repository using ODI Studio. Create a new connection for the primary repository to the new database host, as described in Connecting to the Master Repository in *Developing Integration Projects with Oracle Data Integrator*.
3. Edit each of the Work Repositories. Click **Connection** and edit the connection information so that the JDBC URL points to the new database host containing the work repository.
4. For the Oracle Data Integrator JEE Agent repository configuration, in the Oracle WebLogic Server configuration, edit the data sources to match the new database host location.
5. Update the Oracle Data Integrator Standalone Agent Repository configuration using the following WLST offline commands:

```
cd ORACLE_HOME/odi/common/bin
./wlst.sh
readDomain('DOMAIN_DIRECTORY')
cd('/JdbcSystemResource/OdiMasterRepository/JdbcResource/OdiMasterRepository/
JDBCDriverParams/NO_NAME_0');
set('URL','NEW_JDBC_URL_TO_RECOVERED_DB');
updateDomain();
exit();
```

Recovering Oracle Data Integrator Agents to a Different Host

To recover Oracle Data Integrator agents to a different host:

1. Restore Oracle Data Integrator JEE Agent by restoring the Managed Server, as described in Recovering After Loss of Managed Server Host.
2. Restore the Oracle Data Integrator Standalone system component, as described in Recovering After Loss of Component Host
3. Use ODI Studio to edit each physical agent's configuration and provide the updated Host Name value and, if changed, the Port value.
4. Update Oracle Data Integrator Standalone Agent's host and port configuration using the following commands:

```
cd ORACLE_HOME/odi/common/bin
./wlst.sh
readDomain('DOMAIN_HOME')
```

```
cd('ODI/ODI_STANDALONE_AGENT_NAME')
set("ServerListenAddress",'UPDATED_HOST_NAME');
set("ServerListenPort",'UPDATED_PORT');
updateDomain();
exit();
```

5. Restart the standalone agents and the Oracle Data Integrator applications deployed in Oracle WebLogic Server.

Recovering Oracle WebCenter Content to a Different Host

To recover Oracle WebCenter Content to a different host:

1. If the database must be restored to a different host, restore it, as described in Recovering After Loss of Database Host.
2. Restore the domain, as described in Recovering After Loss of Administration Server Host.
3. If the Vault, WebLayout, or Search directories are not located in the domain directory, restore those directories, if necessary. For example, if the Vault directory is located on a shared drive in /net/home/vault, restore it from backup:

   ```
   cd /net/home/vault
   tar -xf vault_backup_042012.tar
   ```

4. Edit the following file:

   DOMAIN_HOME/ucm_domain/ucm/cs/config/config.cfg

   In the file, change the HttpServerAddress setting to specify the new host. For example:

   ```
   HttpServerAddress=hostname:port_number
   ```

Note that you should restore the database and the shared file system at the same time. If you cannot do that, you can use the IDCAnalyse utility to determine if there are any inconsistencies between the database and the shared file system. If there are, you can perform a manual recovery using IDCAnalyse.

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 topics about each entity may require you to follow one or more of the following procedures. If so, that is noted in the topic describing how to recover the entity.

The following topics describe the actions you may need to take:

- Recovering Fusion Middleware Control to a Different Host
- Modifying the mod_wl_ohs.conf File
- Creating a New Machine for Certain Components
- Updating Oracle Inventory
- Recovering the Windows Registry

Recovering Fusion Middleware Control to a Different Host

To recover Fusion Middleware Control to a different host, update properties using the System MBean Browser:

1. In Fusion Middleware Control, from the WebLogic Domain menu, select **System MBean Browser.**

2. In the System MBean Browser pane, expand **Application-Defined MBeans,** then **emoms.props,** then **Server: AdminServer,** then **Application: em,** and then **Properties.**
3. Click **emoms.properties.**
4. In the Attributes pane, select the Operations tab and click **setProperty.**
5. Change the value of the following properties to the new host name:
   - oracle.sysman.emSDK.svlt.ConsoleServerHost
   - oracle.sysman.emSDK.svlt.ConsoleServerName

   For example, for Key, enter oracle.sysman.emSDK.svlt.ConsoleServerHost. Then, for value, enter host.example.com:7001_Management_Service.
6. Click **Invoke.**

Modifying the mod_wl_ohs.conf File

When you recover an Administration Server or a Managed Server to a different host and your environment includes Oracle HTTP Server, you must modify the following file on the new host:

```
(UNIX) DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_name/mod_wl_ohs.conf
(Windows) DOMAIN_HOME\config\fmwconfig\components\OHS\ohs_name\mod_wl_ohs.conf
```

Note that with Oracle HTTP Server in a WebLogic Server domain, this directory is in the Domain home of the Administration Server. With Oracle HTTP Server in a standalone domain, this directory is the Domain home of Oracle HTTP Server.

Modify all of the instances of the host name, port, and clusters (elements such as WebLogicHost, WebLogicPort, and WebLogicCluster) entries in that file. For example:

```
<Location /console>
    SetHandler weblogic-handler
    WebLogicHost Admin_Host
    WeblogicPort Admin_Port
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>
 .
 .
 .
<Location /soa-infra>
   SetHandler weblogic-handler
   WebLogicCluster SOAHOST1VHN2:8001,*SOAHOST2VHN1*:*8001*
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

Creating a New Machine for Certain Components

If the Administration Server has a Listen address, you must create a new machine with the new host name before you start the Administration Server:

Take the following steps:

1. Create a new machine with the new host name. Use the following WLST commands, in offline mode:

   ```
   readDomain('DOMAIN_HOME')
   machine = create('newhostname', 'Machine')
   cd('/Machine/newhostname')
   ```

```
nm = create('newhostname', 'NodeManager')
cd('/Machine/newhostname/NodeManager/newhostname')
set('ListenAddress', 'newhostname')
updateDomain()
```

2. For the Administration Server, set the machine with the new host name, using the following WLST command, in offline mode:

```
readDomain('DOMAIN_HOME')
cd ('/Machine/newhostname')
machine = cmo
cd ('/Server/AdminServer')
set('Machine', machine)
updateDomain()
```

3. Set the listen port for the Administration Server, using WLST:

```
readDomain('DOMAIN_HOME')
cd('/Server/AdminServer')
cmo.setListenPort(8001)
updateDomain()
```

4. If required, update the Administration Server listen address, using WLST:

```
readDomain('DOMAIN_HOME')
cd('/Server/AdminServer')
cmo.getListenAddress()
cmo.setListenAddress('newhostname')
updateDomain()
exit()
```

Updating Oracle Inventory

For many components, when you recover to a different host, as in the case of loss of host, you must update the Oracle inventory. To do so, execute the following script:

```
(UNIX) ORACLE_HOME/oui/bin/attachHome.sh
(Windows) ORACLE_HOME\oui\bin\attachHome.cmd
```

Recovering the Windows Registry

When you recover any component to a different host on Windows, as in the case of loss of host, you must import any Windows Registry keys related to Oracle Fusion Middleware to the new host. (You exported the Registry keys in Backing Up Windows Registry Entries.)

Recover the following Registry key.

```
HKEY_LOCAL_MACHINE\Software\Oracle
```

In addition, recover each node that begins with **Oracle** within the following registry keys:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services
```

To import a key that you have previously exported, use the following command:

```
regedit  /I  FileName
```

For example:

```
regedit /I C:\oracleregistry.reg
```

You can also use the Registry Editor to import the key, as described in the Registry Editor Help.

Recovering After Loss of Database Host

For information about recovering your database, see Recovering a Database.

Part VIII

Advanced Administration: Expanding Your Environment

You can expand your Oracle Fusion Middleware environment using scaling.

19

Scaling Up Your Environment

You can expand your environment by adding Managed Servers, expanding your domain to include other products, creating a cluster of Managed Servers, creating a standalone domain or system component, and copying existing Oracle homes or domains.

Overview of Scaling Up Your Environment

Scalability is the ability of a system to provide throughput in proportion to, and limited only by, available hardware resources. A scalable system is one that can handle increasing numbers of requests without adversely affecting response time and throughput.

The growth of computational power within one operating environment is called vertical scaling. Horizontal scaling is leveraging multiple systems to work together on a common problem in parallel.

Oracle Fusion Middleware scales both vertically and horizontally.

Oracle Fusion Middleware provides great vertical scalability, allowing you to add more Managed Servers or components to the same host. This is known as **scale up**.

Horizontally, Oracle Fusion Middleware can provide failover capabilities to another host computer. That way, if one computer goes down, your environment can continue to serve the consumers of your deployed applications. This is also known as scaling out or machine scale out. See Scaling Out a Topology in the *High Availability Guide*.

Deploying a high availability system minimizes the time when the system is down (unavailable) and maximizes the time when it is running (available). Oracle Fusion Middleware is designed to provide a wide variety of high availability solutions, ranging from load balancing and basic clustering to providing maximum system availability during catastrophic hardware and software failures.

High availability solutions can be divided into two basic categories: local high availability and disaster recovery. See:

- Introduction to High Availability in *High Availability Guide*
- Overview of Oracle Fusion Middleware Disaster Recovery in *Disaster Recovery Guide*

Extending a Domain to Support Additional Components

When you create an Oracle WebLogic Server domain, you create it using a particular domain template. That template supports a particular component or group of components, such as Oracle WebLogic Server. If you want to add other components, such as Oracle HTTP Server, to that domain, you can extend the domain by creating additional Managed Servers in the domain, using a domain template for the component which you want to add.
When you extend a domain, the domain must be offline.

To extend a domain, you use the Oracle WebLogic Server Configuration Wizard from an Oracle home into which the desired component has been installed. Then, you select the domain that you want to extend and the component you want to add. For detailed information, see Configuring Your WebLogic Domain in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

For example, to extend a domain that initially was created to support Oracle Application Development Framework so that it can now also support Oracle HTTP Server:

1. Use RCU to add any required schemas for the component, as described in Creating Schemas in *Creating Schemas with the Repository Creation Utility*.
2. Install Oracle HTTP Server, as described in About the Oracle HTTP Server Installation in *Installing and Configuring Oracle HTTP Server*.
3. From the Oracle home, invoke the Configuration Wizard, using the following command:

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

   The Configuration Wizard's Welcome screen is displayed.
4. Select **Update an existing domain.**
5. In **Domain Location,** specify the location of the domain.
6. Click **Next.**
7. Select **Update Domain Using Product Templates.**
8. Select **Oracle HTTP Server (colocated).**
9. Click **Next.**
10. Select **Extend my domain automatically to support the following added products,** Then, select the source from which this domain is to be extended. For example, select **Oracle HTTP Server.**
11. Click **Next.**
12. Select either **RCU Data** or **Manual Configuration.** If you select RCU Data, the information is automatically populated when you then select **Get RCU Configuration.** If you select Manual Configuration, click **Next.**

    Select the schemas for the new component you added. If the values in the Component Datasources page are not correct, modify the values.
13. Click **Next.**

    The JDBC Component Schema Test screen is displayed.
14. If the test succeeds, click **Next.**

The Advanced Configuration screen is displayed.

15. Select **System Components.**
16. Click **Next.**
17. Click **Add** to create a new Oracle HTTP Server instance.
18. Enter a name for the instance and select OHS as the component type.
19. Click **Next.**
20. The fields in the OHS Server page are prepopulated.
21. Click **Next.**
22. If you do not want to create a new machine, in the Machines page, click **Next.**
23. In the Assign System Components page, double-click the server to move it under the machine.
24. Review the information on the screen and if it is correct, click **Update.**
25. When the operation completes, click **Done.**

Adding Managed Servers to a Domain

You can add Managed Servers to a domain to increase the capacity of your system. The Managed Servers can be added to a cluster.

When a Managed Server is added to a cluster, it inherits the applications and services that are targeted to the cluster. When a Managed Server is not added as a part of a cluster, it does not automatically inherit the applications and services from the template.

To add a Managed Server to a domain, you can use Fusion Middleware Control, the Oracle WebLogic Remote Console, or WLST.

To add a Managed Server to a domain using the Fusion Middleware Control:

1. From the WebLogic Domain menu, choose **Environment,** then, **Servers.**

   The Servers page is displayed.
2. Click **Create.**

   The Create a Server page is displayed.
3. For **Name,** enter a name for the server.

   Each server within a domain must have a name that is unique for all configuration objects in the domain. Within a domain, each server, computer, cluster, JDBC connection pool, virtual host, and any other resource type must be named uniquely and must not use the same name as the domain.
4. For **Sever Listen Port,** enter the port number from which you want to access the server instance.

   If you run multiple server instances on a single computer, each server must use its own listen port.
5. Specify whether this server is to be a standalone server or a member of an existing cluster:
   - If this server is to be a standalone server, select **No, this is a standalone server.**
   - If this server is to be part of an existing cluster, select **Yes, make this server a member of an existing Cluster.** Then, select the cluster.

- If this server is to be part of a new cluster, click **Create a cluster.**

6. Click **Next.**
7. For **Select a Machine,** either select an existing machine or click **Create new machine.**
8. Click **Create**.
9. If the server or cluster did not have Oracle JRF applied, apply JRF, as described in Applying Oracle JRF Template to a Managed Server or Cluster.

- Applying Oracle JRF Template to a Managed Server or Cluster

Applying Oracle JRF Template to a Managed Server or Cluster

Oracle JRF (Java Required Files) consists of those components not included in the Oracle WebLogic Server installation and that provide common functionality for Oracle business applications and application frameworks.

Oracle 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.
- If you create a server using Fusion Middleware Control, the JRF template is automatically applied.
- When you add a Managed Server to a domain and the Managed Server requires JRF services, but the Managed Server is not part of a cluster, you must apply JRF to the Managed Server.
- When you create a new cluster and the cluster requires JRF, you must apply JRF to the cluster.
- You do not need to apply JRF to Managed Servers that are added by product templates during the template extension process (though you must select JRF in the Configuration Wizard).
- You must restart the server or cluster after you apply JRF.

  Note that if you start the server or cluster using Node Manager (for example, through the Remote Console, which uses Node Manager), you must set the Node Manager property `startScriptEnabled` to `true`. See Configuring Node Manager to Start Managed Servers.

The format of the applyJRF command is:

```
applyJRF(target={server_name | cluster_name | *}, domainDir=domain_path,
        [shouldUpdateDomain= {true | false}])
```

You can use the `applyJRF` command online or offline:

- In online mode, the JRF changes are implicitly activated if you use the `shouldUpdateDomain` option with the value `true` (which is the default.) In online mode, this option calls the online WLST save() and activate() commands.

- In offline mode, you must restart the Administration Server and the Managed Servers or cluster. (In offline mode, if you specify the `shouldUpdateDomain` option with the value `true`, this option calls the WLST updateDomain() command.)

For example, to configure the Managed Server server1 with JRF, use the following command:

```
applyJRF(target='server1', domainDir='DOMAIN_HOME')
```

To configure all Managed servers in the domain with JRF, specify an asterisk (*) as the value of the `target` option.

To configure a cluster with JRF, use the following command:

```
applyJRF(target='cluster1', domainDir='DOMAIN_HOME')
```

For additional information about JRF, see:

- Java Required Files Custom WLST Commands in the *WLST Command Reference for Infrastructure Components*
- Using a Different Version of Spring to use a different version of Spring than that which is supplied with JRF

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 Remote Console, or Fusion Middleware Control. This section describes how to create a cluster using Fusion Middleware Control.

To create a cluster of two Managed Servers, wls_server1 and wls_server2:

1. From the WebLogic Domain menu, choose **Environment,** then, **Clusters.**

   The Clusters page is displayed.

2. Expand**Create.** Then, select either **Cluster** or **Dynamic Cluster.**

   For this example, select **Cluster.**
   The Create a Static 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. Click **Next.**
6. In the Add Servers page, select one or more servers to be added to the cluster. In this scenario, select wls_server1 and wls_server2.
7. Click **Create.**

Now, you have a cluster with two members, wls_server1 and wls_server2.

See Understanding WebLogic Server Clustering in *Administering Clusters for Oracle WebLogic Server* for more information about clusters.

Using Elasticity and Dynamic Clusters for On-Demand Scaling

You can configure elastic scaling for dynamic clusters. Elastic scaling adds or removes dynamic server instances on demand or based on certain conditions.

Elasticity allows you to configure elastic scaling for a dynamic cluster based on either of the following:

- Manually adding or removing a running dynamic server instance from an active dynamic cluster. This is called on-demand scaling. You can perform on-demand scaling using the Fusion Middleware component of Enterprise Manager, the WebLogic Remote Console, or the WebLogic Scripting Tool (WLST).
- Establishing policies that set the conditions under which a dynamic cluster should be scaled up or down and actions that define the scaling operations themselves. When the conditions defined in the scaling policy occur, the corresponding scaling action is triggered automatically.

Dynamic clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your application. A dynamic cluster uses a single server template to define configuration for a specified number of generated (dynamic) server instances. When you create a dynamic cluster, the dynamic servers are preconfigured and automatically generated for you, enabling you to easily scale up the number of server instances in your dynamic cluster when you need additional server capacity.

For more information about elasticity and dynamic clusters, see What is Elasticity? in *Configuring Elasticity in Dynamic Clusters for Oracle WebLogic Server*.

Creating a Standalone Domain and a System Component

You can create a standalone domain for system components, such as Oracle HTTP Server, using the Configuration Wizard as described in Configuring Oracle HTTP Server in a Standalone Domain in *Installing and Configuring Oracle HTTP Server*.

Alternatively, you can use WLST to create a standalone domain that contains a system component, for example, Oracle HTTP Server:

1. Invoke WLST from the following directory:

```
cd ORACLE_HOME/oracle_common/common/bin
./wlst.sh
```

2. Read the standalone domain template. For example, for the Oracle HTTP Server standalone domain template:

```
readTemplate('ORACLE_HOME/ohs/common/templates/wls/ohs_standalone_template.jar')
```

3. Configure Node Manager:

```
cd('/')
create('domainName', 'SecurityConfiguration')
cd('SecurityConfiguration/domain_name')
set('NodeManagerUsername', 'username')
set('NodeManagerPasswordEncrypted', 'password')
setOption('NodeManagerType', 'PerDomainNodeManager')
```

4. The standalone template contains default configuration values. However, you can change those values. For example:

```
cd('/OHS/ohs1')
cmo.setAdminHost('127.0.0.1')
cmo.setAdminPort('7779')
cmo.setListenAddress('localhost')
cmo.setListenPort('7777')
cmo.setSSLListenPort('4443')
```

5. Create the domain. Note that this operation takes some time.

```
writeDomain('domain_dir')
closeTemplate()
```

Creating a System Component Instance in a WebLogic Server Domain

You can create a system component instance, such as Oracle HTTP Server, in a WebLogic Server domain using the Configuration Wizard as described in Configuring Oracle HTTP Server in a WebLogic Server Domain in *Installing and Configuring Oracle HTTP Server*.

Alternatively, you can create a system component instance, for example Oracle HTTP Server in the following ways:

- Using Fusion Middleware Control. For example, to create an Oracle HTTP Server instance, see Creating an Instance by Using Fusion Middleware Control in *Administering Oracle HTTP Server.*
- For Oracle HTTP Server using the WLST createOHSInstance command, as described in Creating an Instance by Using WLST in *Administering Oracle HTTP Server*.
- Using WLST commands, as described in this section.

This section describes how to create a system component instance using WLST commands. It uses Oracle HTTP Server as an example and assumes that you have created a WebLogic Server domain that contains Oracle JRF.

1. Invoke WLST from the following directory:

```
cd ORACLE_HOME/oracle_common/common/bin
./wlst.sh
```

2. Read the domain template and add the template for the system component. The following example shows the Oracle HTTP Server template:

```
readDomain('DOMAIN_HOME')
addTemplate('ORACLE_HOME/ohs/common/templates/wls/ohs_managed_template.jar')
```

3. If you have not already created a machine for the system component, create one:

```
cd('/')
create('ohs_machine', 'Machine')
cd('/Machines/ohs_machine')
create('ohs_machine', 'NodeManager')
cd('NodeManager/ohs_machine')
```

In this case, leave the Node Manager port as it is.

4. Create the system component instance, in this case, Oracle HTTP Server:

```
cd('/')
create('myohs', 'SystemComponent')
cd('/SystemComponent/myohs')
cmo.setComponentType('OHS')
set('Machine', 'ohs_machine')
```

5. Configure the system component instance that you just created. Note that the properties that you set will be different for each type of system component. For example, for Oracle HTTP Server:

```
cd('/OHS/myohs')
cmo.setAdminHost('127.0.0.1')
cmo.setAdminPort('7779')
cmo.setListenPort('7777')
cmo.setSSLListenPort('4443')
```

6. Update the domain:

```
updateDomain()
closeDomain()
```

Part IX
Appendixes

You may need to consult this supplemental information, such as the movement script syntax, and accessibility and troubleshooting information.

This part contains the following appendixes:

A

Copy and Paste Binary Files Scripts

Oracle Fusion Middleware provides scripts that you can use to copy the Oracle Fusion Middleware binaries to another system.
You can use these scripts in conjunction with the chghost utility to change the network configuration of your Oracle Fusion Middleware installation or to move it to another system. See Changing Oracle Fusion Middleware Network Configurations for information about the chghost utility and procedures to change the network configuration or move your Oracle Fusion Middleware environment.

This appendix explains the scripts that you can use to copy the Oracle Fusion Middleware binaries to another system.

- About the Copy and Paste Binary Files Scripts
  The movement scripts copy the binary files of an Oracle home from a source environment and paste them at the target environment.
- Syntax for the Copy and Paste Binary Files Scripts

About the Copy and Paste Binary Files Scripts

The movement scripts copy the binary files of an Oracle home from a source environment and paste them at the target environment.

Use these scripts in conjunction with the procedures described in Changing the Host Name of Oracle Fusion Middleware.

Table A-1 shows the scripts you use to move the binary files in an Oracle home.

Table A-1 Copy and Paste Binary Files Scripts

TO:	Script	See:
Copy the binary files of the source Oracle home	(UNIX) *ORACLE_HOME*/oracle_common/bin/copyBinary.sh (Windows) *ORACLE_HOME*\oracle_common\bin\copyBinary.cmd	copyBinary Script
Apply the copied Oracle home to the target	(UNIX) *ORACLE_HOME*/oracle_common/bin/pasteBinary.sh (Windows) *ORACLE_HOME*\oracle_common\bin\pasteBinary.cmd	pasteBinary Script

To view the help on any of these scripts, use the -help option. For example:

```
./pasteBinary.sh -javaHome /scratch/oracle/jdk17.0.12 -help
```

Note that the help shows the UNIX version of the parameter values. For other platforms, such as Windows, change the parameter values for the platform.

> **Note:**
>
> - For the temporary directory, do not provide a path that contains a space.
> - A Universal Uniform Naming Convention (UNC) path is not supported on Windows. For example, the following is not supported:
>
>   `\\host_name\oracle\java\win64\jdk17\jre\bin\java`

Syntax for the Copy and Paste Binary Files Scripts

The following topics describe the syntax of the copyBinary and pasteBinary scripts. The options are described in the tables that follow the syntax.

- copyBinary Script
- pasteBinary Script

> **Note:**
>
> The value of options must not contain a space. For example, on Windows, you cannot pass the following as a value to the -archiveLoc option:
>
> `C:\tmp\Archive Files`
>
> However, the value of the JavaHome option can contain a space.

- copyBinary Script
- pasteBinary Script

copyBinary Script

Creates an archive file of the source Oracle home by copying the binary files of that Oracle home, including its WebLogic Server home, into the archive file.

The copyBinary script is located in:

```
(UNIX) ORACLE_HOME/oracle_common/bin/copyBinary.sh
(Windows) ORACLE_HOME\oracle_common\bin\copyBinary.cmd
```

The syntax is:

```
copyBinary  [-javaHome path_of_jdk]
            -sourceOracleHomeLoc ORACLE_HOME_location
            -archiveLoc archive_location
            [-ignoreDiskWarning {true | false}]
            [-excludeFilesPattern regex1 [, regex2 ...]
            [-includeDirs dir1 [, dir2 ...]
            [-ignoreDefaultExcludes]
            [-maxArchiveSize integer
            [-silent {true | false}]
```

The following example shows how to create an archive of an Oracle home on Linux:

```
copyBinary.sh -javaHome /scratch/oracle/jdk17.0.12
              -archiveLoc /tmp/oh_copy.jar
              -sourceOracleHomeLoc /scratch/oracle/Oracle_home1
              -includeDirs cfgtoollogs,other
              -excludeFilesPattern /inventory/Actions/.*,/inventory/Clone/.*
              -maxArchiveSize 4096
```

> **Note:**
>
> When you execute the script, you must specify a matching Java home. That is, if the Oracle homes are 64 bit, you must specify a 64-bit Java home. If the Oracle homes are 32 bit, you must specify a 32-bit Java home.

Table A-2 describes the options for the copyBinary script.

Table A-2 Options for the copyBinary Script

Options	Description	Mandatory or Optional
-javaHome	The absolute path of the Java Developer's Kit. The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.	Optional
-sourceOracleHomeLoc	The absolute path of the Oracle home to be archived. You can only specify one Oracle home.	Mandatory
-archiveLoc	The absolute path of the archive location. Use this option to specify the location of the archive file to be created with the copyBinary script. The archive location must not exist. Ensure that the archive location is not within the Oracle home structure.	Mandatory
-ignoreDiskWarning	Specifies whether the operation ignores a warning that there is not enough free space available. The default is false. You may need to use this flag if the target is NFS mounted or is on a different file system, such as Data ONTAP.	Optional

-excludeFilesPattern	Specifies files to be excluded from the archive. You can specify more than one file by separating them with commas. Use the following formats:
	`UNIX: -excludeFilesPattern /inventory/Actions/.*,/inventory/Clone/.*`
	`Windows: -excludeFilesPattern "\\inventory\\Actions\\.*,\\inventory\\Clone\\.*"`
		Optional

-includeDirs	Specifies the directories, besides the Oracle Home, to be included in the archive. You specify the top-level directory; subdirectories will also be included in the archive.	Optional
-ignoreDefaultExcludes	The flag specifies that default files are not excluded by default. If the same files are added to -excludeFilePatterns they will be excluded. (By default, some files, such as log files, are excluded.)	Optional
-maxArchiveSize	Specifies the maximum size of an archive file. As the script executes, if the jar file reaches the maximum size, it creates additional jar files, until it completes the process.	Optional

Table A-2 (Cont.) Options for the copyBinary Script

Options	Description	Mandatory or Optional
-silent	Specifies whether the operation operates silently. That is, it does not prompt for confirmation, which is the default. To specify that it does prompt for confirmation, specify this option with the value of `false`. To continue, you must type `yes`, which is not case sensitive. Typing anything other than yes causes the script to abort.	Optional

pasteBinary Script

Applies the archive to the target destination, by pasting the binary files of the source Oracle home to the target environment. You can apply the archive to the same host or a different host.

The pasteBinary script is located in:

```
(UNIX) ORACLE_HOME/oracle_common/bin/pasteBinary.sh
(Windows) ORACLE_HOME\oracle_common\bin\pasteBinary.cmd
```

The syntax is:

```
pasteBinary [-javaHome path_of_jdk]
            -archiveLoc archive_location
            -targetOracleHomeLoc target_Oracle_Home_location
            [-targetOracleHomeName Oracle_home_name]
            [-ignoreDiskWarning {true | false}]
            [-ignoreJavaVersion {true | false}]
            [-ohAlreadyCloned {true | false}]
            [-prereqConfigLoc path_to_config_files]
            [-entryPoint entrypoint_name]
            [-invPtrLoc Oracle_InventoryLocation]
            [-executeSysPrereqs {true | false}]
            [-silent {true | false}]
```

The following example shows how to apply the archive to the directory /scratch/oracle/Oracle_home_prod, on Linux:

```
pasteBinary.sh -javaHome /scratch/oracle/jdk17.0.12
               -archiveLoc  /tmp/oh_copy.jar
               -targetOracleHomeLoc /scratch/oracle/Oracle_home_prod
               -targetOracleHomeName Oracle_home_prod
```

Table A-3 describes the options for the pasteBinary script.

Table A-3 Options for the pasteBinary Script

Options	Description	Mandatory or Optional
-javaHome	The absolute path of the Java Developer's Kit. The script detects if the operating system is 64 bit and passes the -d64 option to the scripts in the command line.	Optional

Table A-3 (Cont.) Options for the pasteBinary Script

Options	Description	Mandatory or Optional
-archiveLoc	The absolute path of the archive location. Use this option to specify the location of the archive file created with the copyBinary script. The location must exist. This option is mutually exclusive with the -ohAlreadyCloned option.	Mandatory
-targetOracleHomeLoc	The absolute path of the target Oracle home. Ensure that the Oracle home directory does not exist at that location, or if it does, it is an empty directory. Otherwise, the script returns an error message. The -targetOracleHomeLoc cannot be inside another Oracle home.	Mandatory
-targetOracleHomeName	The name for the Oracle home. This name is used to register the Oracle home with Oracle Inventory. Spaces are not allowed in the name.	Optional
-ohAlreadyCloned	A flag specifying that the script reconfigure an already existing Oracle home that was created using a storage-level cloning tool. If this flag is set to true, then the target Oracle home should exist and it should contain Oracle home binaries. Valid values are `true` and `false`. The default is `false`. You cannot use this option when you use the -archiveLoc option.	Optional
-ignoreDiskWarning	A flag specifying whether the operation ignores a warning that there is not enough free space available. The default is false. You may need to use this flag if the target is NFS mounted or is on a different file system, such as Data ONTAP.	Optional
-ignoreJavaVesion	A flag specifying whether the operation ignores the version of Java. The default is false.	Optional
-entryPoint	The name of the entry point.	Optional
-prereqConfigLoc	Specifies the path of the prerequisite configuration files.	Optional
-executeSysPrereqs	Specifies whether the pasteBinary operation checks the prerequisites of the Oracle home. The default is that it checks the prerequisites. To specify that it does not check the prerequisites, specify this option with the value `false`.	Optional

Table A-3 (Cont.) Options for the pasteBinary Script

Options	Description	Mandatory or Optional

-invPtrLoc	On UNIX and Linux, the absolute path to the Oracle Inventory pointer. Use this option if the inventory location is not in the default location, so that the operation can register the Oracle homes with the central Oracle inventory specified in the Oracle Inventory pointer file.
	If the oraInst.loc is not present at default location, you must create this file either at default location as a root user or at any other location as a root or normal user. The following shows an example of the contents of the file:
	`inventory_loc=/scratch/oraInventory`
	`inst_group=dba`
	If the directory specified as the inventory_loc does not exist, the operation will create it.
	You must have write permission to the inventory location.
	On AIX and Linux, the default location is /etc/oraInst.loc. In other UNIX platforms, the default location is /var/opt/oracle/oraInst.loc
	This parameter is only supported on UNIX. On Windows, if you specify this parameter, the script returns an error.
		Optional, if the inventory is in the default location. Otherwise, it is mandatory on Linux.

-silent	Specifies whether the operation operates silently. That is, it does not prompt for confirmation, which is the default. To specify that it does prompt for confirmation, specify this option with the value of `false`. To continue, you must type `yes`, which is not case sensitive. Typing anything other than yes causes the script to abort.	Optional
-ignoreDiskWarning	Specifies whether the operation ignores a warning that there is not enough free space available. The default is false. You may need to use this flag if the target is NFS mounted or is on a different file system, such as Data ONTAP.	Optional

- Executing the pasteBinary Script
  When the source environment is on Host A and the target environment is on Host B, the copyBinary script is executed on Host A. However, when you are executing the pasteBinary script on Host B, the prerequisites and procedure may differ based on the scenario.

Executing the pasteBinary Script

When the source environment is on Host A and the target environment is on Host B, the copyBinary script is executed on Host A. However, when you are executing the pasteBinary script on Host B, the prerequisites and procedure may differ based on the scenario.

This section describes the steps to perform in the following scenarios:

- When the Oracle home does *not* exist on Host B

  Use the `java -jar` option and execute the following command:

```
(UNIX) <JAVA_HOME>/bin/java
-jar <archive_file> -targetOracleHomeLoc <>[-javaHome java_home] [-silent true|
false] [-invPtrLoc orainst_loc_file] [optional arguments]
(Windows) <JAVA_HOME>\bin\java.exe
```

```
-jar <archive_file> -targetOracleHomeLoc <> [-javaHome java_home] [-silent true|
false] [optional arguments]
```

- When executing the pasteBinary script from an existing Oracle home on Host B

> **Note:**
>
> In this scenario, the Oracle home should be previously installed to allow the pasteBinary script to execute successfully. Do not install Oracle Fusion Middleware to create the Oracle home for the pasteBinary purpose only.

1. Ensure that the following (pre-existing) files are available with the same permissions as in the original Oracle home, and in the same structure:

```
%ORACLE_HOME%/oracle_common/bin/pasteBinary.*
%ORACLE_HOME%/oracle_common/modules/*
%ORACLE_HOME%/oui/bin/*
%ORACLE_HOME%/oui/lib/*
%ORACLE_HOME%/oui/modules/*
%ORACLE_HOME%/oui/oraparam.ini
```

2. Execute the following command:

```
(UNIX) ORACLE_HOME/oracle_common/bin/pasteBinary.sh
-archiveLoc <> -targetOracleHomeLoc <> [-javaHome java_home] [-silent true|
false] [-invPtrLoc orainst_loc_file] [optional arguments]
(Windows) ORACLE_HOME\oracle_common\bin\pasteBinary.cmd
-archiveLoc <> -targetOracleHomeLoc <> [-javaHome java_home] [-silent true|
false] [optional arguments]
```

> **Note:**
>
> The `-invPtrLoc` parameter is only supported on UNIX. If you specify this parameter on Windows, then the script returns an error.

See Moving Oracle Fusion Middleware to a New Host for information on how to move Oracle Fusion Middleware to a new host.

B

Oracle Fusion Middleware Command-Line Tools

Oracle Fusion Middleware provides several command-line tools, most of which are located in the Oracle home.
Table B-1 lists the command line tools for Oracle Fusion Middleware.

Table B-1 Oracle Fusion Middleware Command-Line Tools

Command	Path	Description
adrci	UNIX: *ORACLE_HOME*/oracle_common/adr/adcri.sh Windows: *ORACLE_HOME*\oracle_common\adr\adrci.bat	Package incident and problem information into a zip file for transmission to Oracle Support.
bulkdelete	UNIX: *ORACLE_HOME*/ldap/bin/bulkdelete.sh Windows: *ORACLE_HOME*\ldap\bin\bulkdelete.bat	Delete a subtree efficiently inOracle Internet Directory. **See:**bulkdelete in the *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:**bulkload in the *Reference for Oracle Identity Management*
bulkmodify	UNIX: *ORACLE_HOME*/ldap/bin/bulkmodify.sh Windows: *ORACLE_HOME*\ldap\bin\bulkmodify.bat	Modify a large number of existingOracle Internet Directory entries in an efficient way. **See:** bulkload in the *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 inOracle Internet Directory. **See:** catalog in the *Reference for Oracle Identity Management*
chghost	UNIX: *ORACLE_HOME*oracle_common/bin/chghost.sh Windows:*ORACLE_HOME*\oracle_common\bin\chghost.cmd	Changes the host name or IP address of Oracle Fusion Middleware. **See:**About the chghost Utility.
config	UNIX: *ORACLE_HOME*/oracle_common/common/bin/config.sh Windows:*ORACLE_HOME*\oracle_common\common\bin\config.cmd	Invoke the Configuration Wizard to created and configure a domain or extend a domain. **See:** The Installation Guide for the component.
ldapadd	UNIX: *ORACLE_HOME*/bin/ldapadd Windows: *ORACLE_HOME*\bin\ldapadd	Add entries, their object classes, attributes, and values to Oracle Internet Directory. **See:** ldapadd in the *Reference for Oracle Identity Management*

Table B-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:** ldapaddmt in the *Reference for Oracle Identity Management*
ldapbind	UNIX: *ORACLE_HOME*/bin/ldapbind Windows: *ORACLE_HOME*\bin\ldapbind	Add and delete catalog entries inOracle Internet Directory. **See:** ldapbind in the *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 theOracle Internet Directory. **See:** ldapcompare in the *Reference for Oracle Identity Management*
ldapdelete	UNIX: *ORACLE_HOME*/bin/ldapdelete Windows: *ORACLE_HOME*\bin\ldapdelete	Remove entire entries fromOracle Internet Directory. **See:** ldapdelete in the *Reference for Oracle Identity Management*
ldapmoddn	UNIX: *ORACLE_HOME*/bin/ldapmoddn Windows: *ORACLE_HOME*\bin\ldapmoddn	Modify the DN or RDN of anOracle Internet Directory entry. **See:** ldapmoddn in the *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:** ldapmodify in the *Reference for Oracle Identity Management*
ldapmodifymt	UNIX: *ORACLE_HOME*/bin/ldapmodifymt Windows: *ORACLE_HOME*\bin\ldapmodifymt	Modify several Oracle Internet Directory entries concurrently. **See:** ldapmodifymt in the *Reference for Oracle Identity Management*
ldapsearch	UNIX: *ORACLE_HOME*/bin/ldapsearch Windows: *ORACLE_HOME*\bin\ldapsearch	Perform actions on attributes inOracle Internet Directory. **See:** ldapsearch in the *Reference for Oracle Identity Management*
ua	UNIX: *ORACLE_HOME*/oracle_common/upgrade/bin /ua Windows: *ORACLE_HOME*\oracle_common\upgrade\ua.bat	**See:** . • About Using the Oracle Fusion Middleware Upgrade Assistant in *Upgrading with the Upgrade Assistant* • Upgrade Planning Roadmap in *Planning an Upgrade of Oracle Fusion Middleware*
orapki	UNIX: *ORACLE_HOME*/oracle_common/bin/orapki Windows: *ORACLE_HOME*\oracle_common\bin\orapki.bat	Manages wallets and certificates. See orapki.

Table B-1 (Cont.) Oracle Fusion Middleware Command-Line Tools

Command	Path	Description
wlst	UNIX: *ORACLE_HOME*/oracle_common/common/bin/wlst.sh Windows: *ORACLE_HOME*\oracle_common\common\bin\wlst.cmd	(WebLogic Scripting Tool). Manages Oracle WebLogic Server and the components in an Oracle WebLogic Server domain. **See:** • Getting Started Using the Oracle WebLogic Scripting Tool (WLST) • *WLST Command Reference for Oracle WebLogic Server* • WLST Command Reference for Infrastructure Components

C

URLs for Products

You may need to know the URLs needed to access Oracle Fusion Middleware products. Table C-1 shows the URLs to access products after installation.

The URLs in the table are shown with the default ports. The products in your environment might use different ports. To determine the port numbers, from the WebLogic Domain menu in Fusion Middleware Control, select **Port Usage**.

Table C-1 URLs for Products

Product Name	URL with Listen Port	URL with SSL Listen Port	URL with Administration Port
Oracle B2B - console	http://*host*:7003/b2b	https://*host*:7004/b2b	
Oracle Business Activity Monitoring - start page	http://*host*:7001/oracleBAM	https://*host*:7002/oracleBAM	
Oracle Enterprise Manager Fusion Middleware Control	http://*host*:7001/em	https://*host*:7002/em	
Oracle HTTP Server - default access	http://*host*:7777	https://*host*:4443	
Oracle Managed File Transfer - console	http://*host*:7001/mftconsole	https://*host*:7002/mftconsole	
Oracle Service Bus - console	http://*host*:7001/servicebus	https://*host*:7002/servicebus	
Oracle WebLogic Server - WebLogic Hosted Remote Console	http://*host*:7001/rconsole	https://*host*:7002/console	https://*host*:9002/rconsole

D

Port Numbers

Oracle Fusion Middleware allocates servers and port numbers for products and components when you install and configure those products.

- Port Numbers by Product and Component
  Each product or component has one or more default servers and port numbers.

Port Numbers by Product and Component

Each product or component has one or more default servers and port numbers.

This section provides the following information for each Oracle Fusion Middleware product and component Server or Managed Server that uses a port:

- **Product Name:** The name of the product and component.
- **Server or Managed Server Name:** Server or Managed Server Names created in the domain when first configured or created.
- **Listen Port:** The default listen port number Oracle Fusion Middleware attempts to assign to a product's or component's Server or Managed Server when the domain is first configured or created.
- **SSL Listen Port:** The default SSL listen port number Oracle Fusion Middleware attempts to assign to a product's or component's Server or Managed Server when the domain is first configured or created.
- **Administration Port:** The default administration port number Oracle Fusion Middleware attempts to assign to a product's or component's Server or Managed Server when the domain is first configured or created.

Table D-1 shows the default Server or Managed Server and port number for products or components.

Table D-1 Port Numbers Sorted by Component

Product Name	Server/Managed Server Name	Listen Ports	SSL Ports	Administration Ports
Oracle Access Manager	oam_server1	14100	14101	9508
	oam_policy_mgr1	14150	14151	9509
Oracle Business Activity Monitoring (BAM)	bam_server1	7005	7006	9005
Oracle Data Integrator	ODI_server1	15101	15102	9011
Oracle Enterprise Data Quality (EDQ)	edq_server1	8001	7503	9003
Oracle Forms and Reports	WLS_FORMS	9001	9501	9991
	WLS_REPORTS	9012	9512	9992

Table D-1 (Cont.) Port Numbers Sorted by Component

Product Name	Server/Managed Server Name	Listen Ports	SSL Ports	Administration Ports
Oracle Identity Governance	oim_server1	14000	14001	9010
Oracle Internet Directory		3060	3131	NA
Oracle Service Bus (OSB)	osb_server1	8002	8003	9007
Oracle SOA Suite	soa_server1	7003	7004	9004
Oracle Unified Directory		1389	1636	4444
WebCenter Content	UCM_server1	16200	16201	9200
	URM_server1	16300	16301	9300
	IBR_server1	16250	16251	9250
WebCenter Content ADF	WCCADF_server1	16225	16226	9225
WebCenter Imaging	IPM_server1	16000	16001	9600
WebCenter Enterprise Capture	capture_server1	16400	16401	9164
Oracle WebCenter Portal	WC_Portal	8888	8788	9008
	WC_Portlet	8889	8789	9009
Oracle WebCenter Sites	wcsites_server1	7103	7104	9111
	sitecapture_server1	7105	7106	9112
	visitorservices_server1	7107	7108	9113
	satelite_server1	7109	7110	9114
Oracle WebLogic Server	Administration Server	7001	7002	9002

E

Using Oracle Fusion Middleware Accessibility Options

Oracle Fusion Middleware provides accessibility options, such as support for the use Java Access Bridge, more accessible HTML pages and charts, and keyboard navigation.

- Install and Configure Java Access Bridge (Windows Only)
  Java Access Bridge is a technology that exposes the Java Accessibility API in a Microsoft Windows DLL, enabling Java applications and applets that implement the Java Accessibility API to be visible to assistive technologies on Microsoft Windows systems.
- Enabling Fusion Middleware Control Accessibility Mode
  You can make HTML pages of Fusion Middleware Control more accessible and you can more easily view information in charts.
- Fusion Middleware Control Keyboard Navigation

Install and Configure Java Access Bridge (Windows Only)

Java Access Bridge is a technology that exposes the Java Accessibility API in a Microsoft Windows DLL, enabling Java applications and applets that implement the Java Accessibility API to be visible to assistive technologies on Microsoft Windows systems.

If you are installing on a Windows computer, you can install and configure Java Access Bridge for Section 508 Accessibility:

1. Download Java Access Bridge from the following URL:

   `http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136191.html`
2. Install Java Access Bridge.
3. Copy the `access-bridge.jar` and `jaccess-1_4.jar` files from your installation location to the `jre/lib/ext` directory.
4. Copy the `WindowsAccessBridge.dll`, `JavaAccessBridge.dll`, and `JAWTAccessBridge.dll` files from your installation location to the `jre/bin` directory.
5. Copy the `accessibility.properties` file to the `jre/lib` directory.

Enabling Fusion Middleware Control Accessibility Mode

You can make HTML pages of Fusion Middleware Control more accessible and you can more easily view information in charts.

The following topics 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

Making HTML Pages More Accessible

In Fusion Middleware Control, you can enable screen reader support. Screen reader support improves behavior with a screen reader. This is accomplished by adding accessibility-specific constructs to the HTML, and by altering some navigation elements on the pages.

To enable screen reader mode in Fusion Middleware Control:

1. Choose the user name at the right top of the page, then **Accessibility.**

   The Accessibility Preference page is displayed.

2. Select any of the following options:

   - **I use a screen reader:** (Accessibility-specific constructs are added to improve behavior with a screen reader.)
   - **I use high contrast settings:** The fonts use a high contrast.
   - **I use large fonts:** The fonts are larger than normal.
   - **Show me the Accessibility Preference dialog when I log in:** When you log in, the Accessibility Preference page is displayed.

3. Click **OK.**
4. Click Enterprise Manager at the top of the page to return to the page you last visited.

When you select screen reader support, Fusion Middleware Control renders the Web pages so that they can be read by a screen reader. For example, each node in the navigation tree includes a Select button.

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.

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:

Keyboard Navigation for Common Tasks

- 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.

#unique_573/unique_573_Connect_42_CIAFJFAC shows some common tasks and the keyboard navigation used.

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
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

Viewing Release Numbers

Oracle Fusion Middleware release numbers adhere to a specific format.
This appendix describes the format and how to view Oracle Fusion Middleware release numbers.

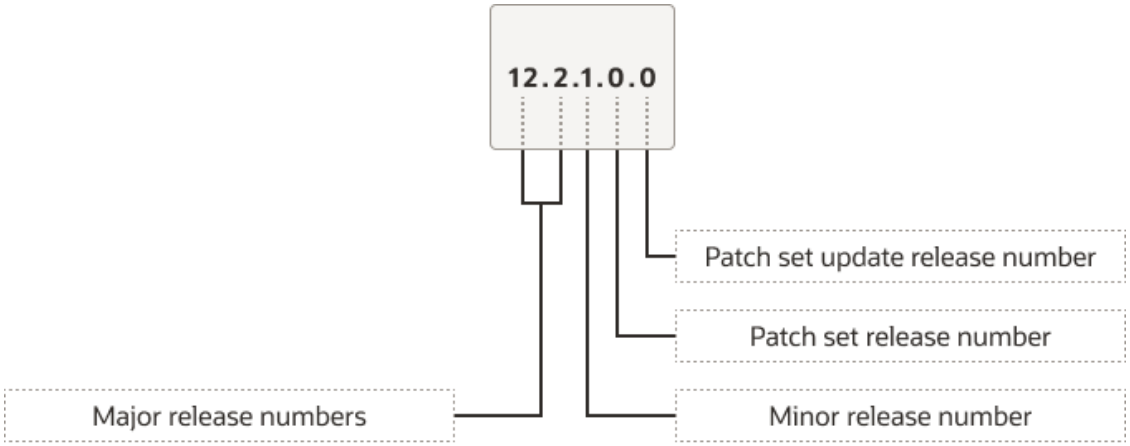
- Release Number Format
  Oracle Fusion Middleware release numbers adhere to a specific format.
- Viewing the Software Inventory and Release Numbers
  All Oracle Fusion Middleware installations and components have a release number.

Release Number Format

Oracle Fusion Middleware release numbers adhere to a specific format.

To understand the release level nomenclature used by Oracle, examine the example of an Oracle Fusion Middleware release number shown in Figure F-1.

Figure F-1 Example of an Oracle Fusion Middleware Release Number



In Figure F-1, each digit is labeled:

- The first two numbers are the Major Release number.

  This is the most general identifier. It represents a major new edition (or version) of Oracle Fusion Middleware, and indicates that the release contains significant new functionality.
- The third number is the Minor release number.
- The fourth number indicates a Patch Set release.
- The fifth number indicates a Patch Set Update release.

Viewing the Software Inventory and Release Numbers

All Oracle Fusion Middleware installations and components have a release number.

The following topics describe how to obtain the release numbers of Oracle Fusion Middleware and its components:

- Viewing Oracle Fusion Middleware Installation Release Numbers
- Viewing Oracle WebLogic Server Release Numbers
- Viewing Component Release Numbers
- Viewing Oracle Internet Directory Release Numbers
- Viewing Metadata Repository Release Numbers
- Viewing Schema Release Numbers

Viewing Oracle Fusion Middleware Installation Release Numbers

All Oracle Fusion Middleware installations have a release number. This number is updated when you apply a patch set release or upgrade the installation.

You can view the release number of an Oracle Fusion Middleware installation using Opatch. Run the following command:

```
(UNIX) ORACLE_HOME/OPatch/opatch lsinventory
(Windows) ORACLE_HOME\OPatch\opatch lsinventory
```

For example, on UNIX:

```
 ./opatch lsinventory
Copyright (c) 2014, Oracle Corporation.  All rights reserved.


Oracle Home       : /scratch/oracle1/Oracle/Middleware/Oracle_Home
Central Inventory : /scratch/oracle1/oraInventory
   from           : /scratch/oracle1/Oracle/Middleware/Oracle_Home/oraInst.loc
OPatch version    : 13.2.0.0.0
OUI version       : 13.2.0.0.0
Log file location : /scratch/oracle1/Oracle/Middleware/Oracle_Home/cfgtoollogs/opatch/
opatch2014-05-29_13-23-02PM_1.log


OPatch detects the Middleware Home as "/scratch/oracle1/Oracle/Middleware/Oracle_Home"
May 29, 2014 1:23:33 PM oracle.sysman.oii.oiii.OiiiInstallAreaControl initAreaControl
INFO: Install area Control created with access level  0
Lsinventory Output file location : /scratch/oracle1/Oracle/Middleware/Oracle_Home/
cfgtoollogs/opatch/lsinv/lsinventory2014-05-29_13-23-02PM.txt

--------------------------------------------------------------------------------

There are no Interim patches installed in this Oracle Home.
```

Viewing Oracle WebLogic Server Release Numbers

You can use the following command to view the release number of Oracle WebLogic Server:

```
(UNIX) java -cp $WL_HOME/server/lib/weblogic.jar weblogic.version
```

```
(Windows) java -cp %WL_HOME%\server\lib\weblogic.jar weblogic.version
```

For example, on UNIX:

```
java -cp $WL_HOME/server/lib/weblogic.jar weblogic.version
WebLogic Server 12.2.1.4.0 Thu Sep 12 04:04:29 GMT 2019 1974621
```

For example, on Windows:

```
java -cp %WL_HOME%\server\lib\weblogic.jar weblogic.version
WebLogic Server 12.2.1.4.0 Thu Sep 12 04:04:29 GMT 2019 1974621
```

> **Note:**
>
> Use `weblogic.version -verbose` to get subsystem information.
>
> Use `weblogic.utils.Versions` to get version information for all modules.

Viewing Component Release Numbers

All Oracle Fusion Middleware components have a release number and many contain services that have release numbers. These numbers *may* be updated when you apply a patch set release or upgrade the installation.

You can view the release number of components and their services by using the following commands:

- On UNIX:

  ```
  cd ORACLE_HOME/inventory
  ls -d Components*/*/*
  ```

- On Windows:

  ```
  cd ORACLE_HOME/inventory/Componentsn
  dir /S /A:D
  ```

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
- Viewing the Oracle Internet Directory Schema
- Viewing the Oracle Internet Directory Context Versions

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

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 is in this form:

```
cn=BASE,cn=OracleSchemaVersion
orclproductversion=90500
```

Viewing the Oracle Internet Directory Context Versions

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 is in this form:

```
cn=oraclecontext
orclversion=101200
```

Viewing Metadata Repository Release Numbers

If you are using an Oracle Database instance for your metadata repository, you can view the release number of the database using SQL*Plus as follows (you can be connected to the database as any user to issue these commands):

```
SQL> COL PRODUCT FORMAT A40
SQL> COL VERSION FORMAT A15
SQL> COL STATUS FORMAT A15
SQL> SELECT * FROM PRODUCT_COMPONENT_VERSION;

PRODUCT                                  VERSION         STATUS
---------------------------------------- --------------- ----------------
NLSRTL                                   11.2.0.4.0      Production
Oracle Database 11g Enterprise Edition   11.2.0.4.0      Production
```

```
PL/SQL                                    11.2.0.4.0      Production
TNS for Linux:                            11.2.0.4.0      Production
```

Viewing Schema Release Numbers

If you are using an Oracle Database instance for your metadata repository, you can view the release number of the schema using SQL*Plus, as follows:

```
SQL> COL COMP_ID FORMAT A20
SQL> COL COMP_NAME A40
SQL> COL VERSION FORMAT A20
SQL> SELECT COMP_ID, COMP_NAME, VERSION FROM SCHEMA_VERSION_REGISTRY;

COMP_ID             COMP_NAME                                VERSION
--------------------------------------------------------------------------------
 .
 .
 .
MDS                 Metadata Services                         12.1.3.0.0
OPSS                Oracle Platform Security Services         12.1.3.0.0
SOAINFRA            SOA Infrastructure Services               12.1.3.0.0
STB                 Service Table                             12.1.3.0.0
 .
 .
 .
```

G
orapki

The `orapki` utility is a command-line tool to manage certificate revocation lists (CRLs), create and manage Oracle wallets, and create signed certificates for testing purposes. It also provided the SSL Configuration Tool.

Oracle Fusion Middleware provides both command-line (the `orapki` utility) and graphical user interfaces to configure SSL. The Oracle WebLogic Scripting Tool (WLST) and Oracle Enterprise Manager Fusion Middleware Control enable you to manage KSS- and JKS-based keystores, wallets, and certificates.

Topic:

- Using the orapki Utility for Certificate and CRL Management

> **See Also:**
>
> - Doc ID 1629906.1 "How To Create a Wallet via ORAPKI in Fusion Middleware 12c" in the Oracle Technology Network Knowledge Base for additional information and examples of the `orapki` commands shown in this appendix.
> - *WLST Command Reference for Infrastructure Security* for examples of the WLST commands shown in this appendix.
> - Configuring SSL in Oracle Fusion Middleware for details about keystore and wallet management in Oracle Fusion Middleware.

> **Note:**
>
> The `orapki` utility is located in the bin directory of Oracle Home, that is, `$ORACLE_HOME/bin`.

- Using the orapki Utility for Certificate and CRL Management
  You can use the orapki utility to perform some of the basic operations like creating a wallet or creating a certificate.

Using the orapki Utility for Certificate and CRL Management

You can use the orapki utility to perform some of the basic operations like creating a wallet or creating a certificate.

This section contains these topics:

- 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.

- Displaying orapki Help
  You can display all the `orapki` commands that are available for a specific mode.
- Creating Signed Certificates for Testing Purposes
  The `orapki` command-line utility provides a convenient, lightweight way to create signed certificates for testing purposes.
- Managing Oracle Wallets with the orapki Utility
  You can use these `orapki` utility `wallet` module commands in scripts to automate the wallet creation process.
- Managing Certificate Revocation Lists with orapki Utility
  Certificate Revocation Lists (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.
- orapki Utility Commands Summary
  Review the purpose and syntax of these `orapki` commands for managing wallets, certificates and certificate revocation lists.

orapki Overview

The `orapki` utility is provided to manage public key infrastructure (PKI) elements, such as wallets and certificate revocation lists, on the command line so the tasks it performs can be incorporated into scripts. This enables you to automate many of the routine tasks of maintaining a PKI.

This command-line utility can be used to perform the following tasks:

- Creating signed certificates for testing purposes
- Managing Oracle wallets:
  - Creating and displaying Oracle wallets
  - Adding and removing certificate requests
  - Adding and removing certificates
  - Adding and removing trusted certificates
- Managing certificate revocation lists (CRLs):
  - Renaming CRLs with a hash value for certificate validation

`orapki` allows you to import certificates in both DER and PEM formats.

- orapki Syntax
- Environment Setup for orapki

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
```

DN Syntax is Platform-specific

Many `orapki` commands require the specification of the `DN`. On UNIX, the `user_dn` is surrounded by single quotes, for example:

```
$ORACLE_HOME/bin/orapki wallet add
-wallet $ORACLE_HOME/wallet
-dn 'CN=server.example.com, OU=Support, O=Oracle, L=Jaipur, ST=Rajasthan, C=IN'
-keysize 1024
```

Windows requires double quotes:

```
$ORACLE_HOME/bin/orapki wallet add
-wallet $ORACLE_HOME/wallet
-dn "CN=server.example.com, OU=Support, O=Oracle, L=Jaipur, ST=Rajasthan, C=IN"
-keysize 1024
```

Environment Setup for orapki

When running orapki in the context of Web Tier installations, set `ORACLE_HOME` to point to the product installation location.

Displaying orapki Help

You can display all the `orapki` commands that are available for a specific mode.

```
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.

Creating Signed Certificates for Testing Purposes

The `orapki` 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.

Managing Oracle Wallets with the orapki Utility

You can use these `orapki` utility `wallet` module commands in scripts to automate the wallet creation process.

The following topics describe the syntax used to create and manage Oracle wallets with the `orapki` command-line utility:

- Creating and Viewing Oracle Wallets with orapki
- Adding Certificates and Certificate Requests to Oracle Wallets with orapki
- Adding an ECC Certificate to an Oracle Wallet with orapki
- Exporting Certificates and Certificate Requests from Oracle Wallets with orapki
- Creating and Managing Trust Flags
- Importing PKCS#12 Files to an Oracle Wallet
- Converting Between Oracle Wallet and JKS Keystore

Creating and Viewing Oracle Wallets with orapki

This section contains these topics:

- Creating an Oracle Wallet
- Creating an Oracle Wallet with Auto-login Enabled
- Creating an Oracle Wallet with AES Encryption
- Converting an Existing Wallet to Use AES Encryption
- Viewing an Oracle Wallet

Creating an Oracle Wallet

```
orapki wallet create -wallet wallet_location
```

This command prompts you to enter and re-enter a wallet password. It creates a wallet in the location specified for `-wallet`.

Creating an Oracle Wallet with Auto-login Enabled

```
orapki wallet create -wallet wallet_location -auto_login
```

This command creates a wallet with auto-login enabled. 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`.

Creating an Oracle Wallet with AES Encryption

```
orapki wallet create -wallet wallet -pwd pwd -compat_v12
```

This command creates an Oracle wallet with AES encryption.

Converting an Existing Wallet to Use AES Encryption

```
orapki wallet convert -wallet wallet -compat_v12 -pwd pwd
```

This command converts an Oracle wallet from 3DES to AES encryption.

Viewing an Oracle Wallet

```
orapki wallet display -wallet wallet_location
```

This command displays the certificate requests, user certificates, and trusted certificates contained in the wallet.

Adding Certificates and Certificate Requests to Oracle Wallets with orapki

This section contains these topics:

- Adding a Certificate Request to an Oracle Wallet
- Adding a Trusted Certificate to an Oracle Wallet
- Adding a Root Certificate to an Oracle Wallet
- Adding a User Certificate to an Oracle Wallet

Adding a Certificate Request to an Oracle Wallet

```
orapki wallet add -wallet wallet_location -dn user_dn -keysize certificate_key_size -
addext_ski -addext_ku extension_key_usage -addext_basic_cons CA -pathLen number -
addext_san DNS
```

This command adds a certificate request to a wallet for the user with the specified distinguished name (`user_dn`). The request also specifies the following parameters and extensions:

- The `-keysize` parameter specifies the requested certificate's key size. The key size identifiers are `512`, `1024`, `2048`, `4096`, `8192`, `16384`.
- The `-addext_ski` parameter is an extension for adding a subject key identifier extension to a certificate request.

- The `-addext_ku` parameter is an extension for adding key usages. The keys are `digitalSignature`, `nonRepudiation`, `keyEncipherment`, `dataEncipherment`, `keyAgreement`, `keyCertSign`, `cRLSign`, `encipherOnly`, and `decipherOnly`.
- The `-addext_basic_cons` parameter is an extension for adding basic constraints. This extension mentions that the certificate request is CA. It also mentions the `-pathLen`, which signifies the number of non-self-issued intermediate CA certificates that may follow in a valid certification path under CA.
- The `-addext_san` parameter is an extension to X509 certificates used to add subject alternative names, which is used in addition to identity a subject. This option only allows adding domain names separated by a comma. It can be added as shown below in example.

```
-addext_san DNS:<value1>,DNS:<value2>,DNS:<value3>
or
-addext_san DNS:ns1.example.com,DNS:ns2.example.com
```

> **Note:**
>
> The `-addext_san` support is applicable in Oracle Fusion Middleware since 12c (12.2.1.1) and newer.

To sign the request, export it with the export option. See Exporting Certificates and Certificate Requests from Oracle Wallets with orapki.

For example:

```
Linux/Unix:
$ORACLE_HOME/bin/orapki wallet add
-wallet $ORACLE_HOME/wallet
-dn 'CN=server.in.test.com, OU=Support, O=Oracle, L=Jaipur, ST=Rajasthan, C=IN'
-keysize 1024

Windows:
$ORACLE_HOME/bin/orapki wallet add
-wallet $ORACLE_HOME/wallet
-dn "CN=server.in.test.com, OU=Support, O=Oracle, L=Jaipur, ST=Rajasthan, C=IN"
-keysize 1024
```

Adding 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.

Adding a Root Certificate to an Oracle Wallet

```
orapki wallet add -wallet wallet_location -dn
certificate_dn -keysize 512|1024|2048|4096|8192|16384 -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, 4096, 8192 or 16384 bits.

See Adding a Certificate Request to an Oracle Wallet for an example showing the DN syntax.

Adding 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`*.

Adding an ECC Certificate to an Oracle Wallet with orapki

This section contains these topics:

- Adding an ECC Certificate Request to an Oracle Wallet
- Viewing an Oracle Wallet with ECC Certificate

Adding an ECC Certificate Request to an Oracle Wallet

```
orapki wallet add -wallet wallet_location -dn user_dn -sign_alg signing_alg -asym_alg
ECC -eccurve curve_type -addext_ski -addext_ku extension_key_usage -addext_basic_cons CA
-pathLen number -addext_san DNS
```

This command adds a certificate request to a wallet for the user with the specified distinguished name (`user_dn`). The request specifies the following ECC specific parameters:

- The `-sign_alg` parameter specifies the signature algorithm that can be used by CA to sign the certificate. `ecdsasha1`, `ecdsasha256`, `ecdsasha384` and `ecdsasha512` are the supported signing algorithms for CAs having ECC key and `md5`, `sha1`, `sha256`, `sha384` and `sha512` are the signing algorithms supported for CAs with RSA keys.
- The `-asym_alg` parameter specifies the type of key: ECC or RSA. If the key type is ECC, then option `-eccurve` has to be specified to set the ECC curve on which key is generated. If the key type is RSA, then option `-keysize` has to be specified to set the key size of RSA key to be generated.
- The `-eccurve` parameter specifies the curve on which ECC key is generated. The curve identifiers are `p192`, `p224`, `p256`, `p384`, `p521`, `k163`, `k233`, `k283`, `k409`, `k571`, `b163`, `b233`, `b283`, `b409`, `b571`.

Viewing an Oracle Wallet with ECC Certificate

```
orapki wallet display -wallet wallet_location
```

This command displays the ECC certificates contained in the wallet.

Exporting Certificates and Certificate Requests from Oracle Wallets with orapki

This section contains these topics:

- Exporting a Certificate from an Oracle Wallet
- Exporting a Certificate Request from an Oracle Wallet

Exporting a Certificate from an Oracle Wallet

```
orapki wallet export -wallet wallet_location -dn
certificate_dn -cert certificate_filename -issuer_dn dn_of_issuer -serial_num
serial_number_of_certificate
```

This command exports a certificate with the subject's distinguished name (`-dn`) from a wallet to a file that is specified by `-cert`. The command uses the following options to uniquely identify a certificate in a wallet:

- The `-issuer_dn` option specifies the DN of the certificate issuer.
- The `-serial_num` option is used as an identification number for a certificate. The serial number option supports both decimal and hexadecimal format.

See Adding a Certificate Request to an Oracle Wallet for an example showing the DN syntax.

Exporting 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`.

See Adding a Certificate Request to an Oracle Wallet for an example showing the DN syntax.

Creating and Managing Trust Flags

Trust Flags in Oracle Wallet Certificates

Trust flags allow adequate roles to be assigned to certificates to facilitate operations like certificate chain validation and path building. By default, wallets do not support trust flags.

You can use the `orapki` utility to maintain trust flags in the certificates installed in an Oracle Wallet. You can create and convert wallets to support trust flags, create and maintain appropriate flags in each certificate, and so on.

#unique_595/unique_595_Connect_42_CFAFGHDB shows the supported trust flags:

NZ Trust Flag Value	Description and Best Practices	NSS Flag
`SERVER_AUTH` (Trusted server CA certificate)	Assigned to trusted CA's root and intermediate certificates. Useful for fine-grain control to allow CA certificates to act in client CA or server CA roles. Can co-exist with "CLIENT_AUTH" flag. During server authentication, if the server's certificate chain has a CA certificate with a `SERVER_AUTH` flag in the client's certificate store, authentication succeeds. If a CA certificate with a `SERVER_AUTH` flag is not present, authentication fails. In client wallet, assigning the "SERVER_AUTH" flag to server's Root CA certificate is recommended. Server certificate chain verification stops at the certificate with this "SERVER_AUTH" flag. If you do not want to add server's ROOT CA certificate to the client wallet, set it to the server's intermediate CA certificate.	"C"

NZ Trust Flag Value	Description and Best Practices	NSS Flag
CLIENT_AUTH (Trusted client CA certificate)	Assigned to trusted CA's root and intermediate certificates. Can co-exist with the "SERVER_AUTH" flag.Useful for fine-grain control to allow CA certificates to act in client CA or server CA roles. When the SSL server requests client authentication, the server sends a list of subject names of trusted CA certificates it is willing to accept certificates from. Trusted certificates in wallets with the CLIENT_AUTH flag would be used to make this list. During SSL client authentication, if the client's certificate chain has a CA certificate having the CLIENT_AUTH flag in server's wallet, then authentication succeeds. If a CA certificate with the CLIENT_AUTH flag is not present, then authentication fails.	"T"
VALID_PEER	Assigned to peer's user certificate to authenticate peer. Usually it would be without a private key. Cannot co-exist with "CLIENT_AUTH", "SERVER_AUTH", or "USER_CERT" flags. Adding this flag to self-signed server or client certificates is recommended. Certificate chain building and verification stops at the certificate with the "VALID_PEER" flag. During authentication, if the user's certificate sent by an SSL peer for authentication exists in relying party's certificate store with VALID_PEER flag, then this certificate is allowed to establish the SSL connection without any certificate chain validation provided that it is a valid peer's user certificate.	"P"
NULL.	Assigned implicitly to certificates that do not have any flag. Recommended for use with intermediate CA certificates.	""

In addition to the flag assignments you can explicitly perform, here are certain assignments automatically made in certificates when the wallet allows trust flags:

- In a root wallet (with copies of the same certificate in 'user certificates' and 'trusted certificates' section), USER_CERT flag is added to certificate(s) in 'user certificates' section only.
- When a wallet is converted so that it supports trust flags, specific rules govern the assignment of trust flags to the trusted certificates added to the wallet (see Assigning Trust Flags to Trusted Certificates below).
- When a certificate is deleted from the wallet, all flags associated with the certificate are deleted. If the same certificate is re-installed flags must be added again.
- When a wallet is created with trust flags (using the -with_trust_flags option) the wallet is populated with certain default certificates. All these certificates are assigned the SERVER_AUTH/CLIENT_AUTH flags.

Assigning Trust Flags to Trusted Certificates

When you add trusted certificates to wallets which are trust flag-enabled, trust flags are computed as follows:

- Root CA is assigned the SERVER_AUTH flag.
- Intermediate CA (ICA) is assigned the NULL flag.

- End-entity certificate without private key is assigned the VALID_PEER flag.
- Self-signed certificates without private key are assigned the VALID_PEER flag.

The following topics explain the trust flag operations you can perform with orapki:

- Creating a Wallet to Support Trust Flags
- Converting a Wallet to Support Trust Flags
- Adding and Updating a Certificate's Trust Flags
- Adding a Certificate with Trust Flags to Wallet

Creating a Wallet to Support Trust Flags

Use the `orapki` option `with_trust_flags` when creating the wallet.

```
orapki wallet create -wallet wallet_location
-pwd password -with_trust_flags
```

This command creates an Oracle wallet that supports trust flags; wallets created without the `with_trust_flags` parameter do not support trust flags, but can be converted to do so.

Other options like creating an auto-login wallet can also be specified when creating a wallet to support trust flags.

Rules governing the assignment of trust flags to trusted certificates added to a trust-flag-enabled wallet are explained in Creating and Managing Trust Flags (see Assigning Trust Flags to Trusted Certificates), and you can clear these flags explicitly.

Converting a Wallet to Support Trust Flags

You can update an existing wallet to support trust flags.

This command syntax converts a wallet to support trust flags.

```
orapki wallet enable_trust_flags -wallet wallet_location -pwd password
```

or, for auto-login wallet:

```
orapki wallet enable_trust_flags -wallet wallet_location -auto_login_only
```

Usage rules are as follows:

- Password is not required if it is an auto-login wallet.
- After using this command, you cannot convert the wallet back to its original state, that is, to not support trust flags.
- All user certificates present in the wallet are assigned the `USER_CERT` flag.

Trust flags for trusted certificates are computed as follows:

- Root CA is assigned SERVER_AUTH flag.
- ICA or intermediate CA is assigned NULL flag.
- End-entity certificate without private key is assigned VALID_PEER flag.

You can change the flags associated with trusted certificates to assign the desired trust flags to these certificates.

Adding Certificates to Empty Wallet

As mentioned earlier, after using this command you cannot convert the wallet back to its original state to not support trust flags.

If you remove all the certificates from the wallet, including the default certificates installed by orapki, the tool can no longer determine whether the wallet supports trust flags. Therefore it is advisable not to remove the default installed certificates from the wallet; if you must remove them, make sure to install a certificate before removing them so at least one certificate remains in the wallet.

If you delete all the certificates from a wallet and later install new certificates, the wallet behaves as follows: If the new certificate is installed with the trust flags option, the wallet will automatically support trust flags. If the new certificate is installed without the trust flags option, the wallet will not support trust flags.

Adding and Updating a Certificate's Trust Flags

The `orapki` option `trust_flags` assigns the requisite flags to selected certificates.

```
orapki wallet assign_trust_flags -wallet wallet_location
-pwd password -trust_flags ""|"flags"
-dn "value" [-serial_num "value"  -issuer "value"]
```

This command adds, updates, or deletes trust flags for the certificate specified by the `dn`. Syntax rules are as follows:

- The wallet must support trust flags.
- Password is not required if wallet is an auto-login wallet.
- Specify the flags as defined in #unique_595/unique_595_Connect_42_CFAFGHDB.
- The Subject DN is the only mandatory certificate attribute parameter, the remaining two parameters being optional. However, you must provide sufficient detail using these parameters to uniquely identify the certificate.
- The matching attribute names are case insensitive, and attribute values are case-sensitive.
- The `-serial_num` option is used as an identification number for a certificate. The serial number option supports both decimal and hexadecimal format.
- Existing flags, if any, assigned to the certificate are over-written.
- Multiple flags can be assigned using ", "(comma); like –add "`SERVER_AUTH,CLIENT_AUTH`"
- The `USER_CERT` flag is not permitted in this command, as this flag is assigned implicitly to the user certificates. for the user certificate the USER_CERT flag shall always be there.
- To remove trust flags, use `-add ""`. The NULL flag is assigned to the certificate.
- if the modify/clear action would result in an invalid certificate chain for any user certificate, the action is not carried out.

For example:

```
orapki wallet assign_trust_flags -wallet /usr/test
-trust_flags "SERVER_AUTH,CLIENT_AUTH"
-dn "cn=jack, ou=people, dc=example, dc=com"
-serial_num "1122" -issuer "sample"
```

Adding a Certificate with Trust Flags to Wallet

Use the `orapki` option `trust_flags` when adding certificates to a wallet.

```
orapki wallet add -wallet wallet_location
-[trusted_cert|user_cert|self_signed]
-cert cert_location -pwd password -trust_flags "flag(s)"
```

This command adds a certificate with specified trust flag(s) to an Oracle wallet. Syntax rules are as follows:

- The wallet must support trust flags.
- Passwords are not required if the wallet is an auto-login wallet.
- `cert_location` is not required if you generate a self signed certificate.
- `USER_CERT` flag is added implicitly if the certificate is of type `user_cert`. (In a root wallet a self-signed certificate is also present in the 'trusted certificates' section; the USER_CERT flag is *not* assigned to this certificate).
- The flags are specified as defined in #unique_595/unique_595_Connect_42_CFAFGHDB.
- If trust flags are enabled there is no need for the complete hierarchy of trusted certificates to be present (unlike the case for wallets without trust flags, where the entire chain must be present when adding a user certificate). The certificate chain building stops if a `SERVER_AUTH/CLIENT_AUTH` flag is assigned to any trusted certificate in the hierarchy.

Importing PKCS#12 Files to an Oracle Wallet

The `orapki` option `pkcs12file` enables you to import PKCS#12 files into a wall.et

```
orapki wallet import_pkcs12
-wallet wallet_location [-pwd wallet_password]
-pkcs12file pkcs12_file_location [-pkcs12pwd pkcs12_file_password]
```

This command imports a PKCS#12 file into an Oracle wallet. The utility prompts you if you do not specify passwords with the command.

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
- Converting 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 password -jksKeyStoreLoc ./ewallet.jks -
jksKeyStorepwd password2
```

This example migrates keys and trusted certificate entries into separate JKS keystores:

```
orapki wallet pkcs12_to_jks -wallet ./ -pwd password1
-jksKeyStoreLoc ./ewalletK.jks -jksKeyStorepwd password2
-jksTrustStoreLoc ./ewalletT.jks -jksTrustStorepwd password2
```

Managing Certificate Revocation Lists with orapki Utility

Certificate Revocation Lists (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.

> **See Also:**
>
> "Certificate Revocation List Management" in the *Oracle Database Advanced Security Administrator's Guide* for details about managing CRLs with `orapki`.

The following sections describe CRLs, how you use them, and how to use orapki to manage them:

- About Certificate Validation with Certificate Revocation Lists
- Certificate Revocation List Management

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.

- What CRLs Should You Use?
- How CRL Checking Works

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.

How CRL Checking Works

Certificate revocation status is checked against CRLs which are located in file system directories, or downloaded from the location specified in the CRL Distribution Point (CRL DP) extension on the certificate. If you store your CRLs on the local file system or in the directory, then you must update them regularly. If you use CRL DPs then CRLs are downloaded when the corresponding certificates are first used.

The server searches for CRLs in the following locations in the order listed. When the system finds a CRL that matches the certificate CA's DN, it stops searching.

1. Local file system

   The locations and management of CRL files is component-dependent. For Oracle WebLogic Server, see "Configuring the CRL Local Cache" in *Administering Security for Oracle WebLogic Server*. For Oracle HTTP Server, see Doc ID 1665286.1, "How to Configure CRL Checking in Oracle HTTP Server in FMW 12c" in the Oracle Technology Network Knowledge Base.

2. CRL DP

   If the CA specifies a location in the CRL DP X.509, version 3, certificate extension when the certificate is issued, then the appropriate CRL that contains revocation information for that certificate is downloaded. Currently, Oracle Advanced Security supports downloading CRLs over HTTP and LDAP.

> **Note:**
>
> - For performance reasons, only user certificates are checked.
> - Oracle recommends that you store CRLs in the directory rather than the local file system.

Certificate Revocation List Management

Procedures for CRL management depend on the component in question. For Oracle WebLogic Server, see Configuring the CRL Local Cache in *Administering Security for Oracle WebLogic Server*. For Oracle HTTP Server, see Doc ID 1665286.1, "How to Configure CRL Checking in Oracle HTTP Server in FMW 12c" in the Oracle Technology Network Knowledge Base.

Before you can enable certificate revocation status checking, you must ensure that the CRLs you receive from the CAs you use are in a form (renamed with a hash value) or in a location (uploaded to the directory) in which your system can use them. Oracle Advanced Security provides a command-line utility, `orapki`, that you can use to perform the following task:

> **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.

- Renaming CRLs with a Hash Value for Certificate Validation

> **See Also:**
>
> Command-Line Tools Overview in the *Oracle Fusion Middleware Reference for Oracle Identity Management* for information about LDAP command-line tools and their syntax.

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.

orapki Utility Commands Summary

Review the purpose and syntax of these `orapki` commands for managing wallets, certificates and certificate revocation lists.

- orapki cert create
- orapki cert display
- orapki crl create
- orapki crl hash
- orapki crl revoke
- orapki crl status
- orapki crl verify
- orapki wallet add
- orapki wallet change_pwd

- orapki wallet create
- orapki wallet enable_trust_flags
- orapki wallet assign_trust_flags
- orapki wallet display
- orapki wallet export
- orapki wallet export_trust_chain
- orapki wallet import_pkcs12

orapki cert create

Use this command to create a signed certificate for testing purposes.

The syntax for FMW is:

```
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.

orapki cert display

Use this command to display details of a specific certificate.

orapki crl create

The syntax for FMW to create a CRL is:

```
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

The syntax for ENT to create a CRL is:

```
crl:
create [-crl [url|filename]] [-wallet [cawallet]] <-issuer [issuer_dn]>
<-issuersissuer [issuersissuer_dn]>
<-serial_num [serial_num]> [-nextupdate [days]] [-pwd <pwd>] [-sign_alg
<md5|sha1|sha256|sha384|sha512|ecdsasha1|ecdsasha256|ecdsasha384|ecdsasha512>]
```

- -crl is the location where the CRL will be created (for example `./nzcrl.txt`)
- -issuer is the DN of the issuer
- -issuersissuer is the issuer DN of the issuer certificate
- -serial_num is the serial number for the CRL
- -sign_alg is the sign algorithm to be used

orapki crl hash

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.

The syntax is:

```
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.

orapki crl revoke

Use these commands to revoke a certificate.

The syntax for FMW is:

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

The syntax for ENT is:

```
revoke [-crl [url|filename]] [-wallet [cawallet]] [-cert [revokecert]] [-pwd
<pwd>] [-sign_alg
<md5|sha1|sha256|sha384|sha512|ecdsasha1|ecdsasha256|ecdsasha384|ecdsasha512>]
```

- `-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.
- `-sign_alg` is the sign algorithm to be used.

orapki crl status

Use this command to check if a certificate is revoked in a CRL.

The syntax is:

```
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

orapki crl verify

Use this command to verify a CRL signature.

The syntax is:

```
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

orapki wallet add

Use this command to add certificate requests and certificates to an Oracle wallet.

> **See Also:**
>
> See Adding a Certificate Request to an Oracle Wallet for an example showing the DN syntax.

To add certificate requests:

```
orapki wallet add -wallet wallet_location -dn user_dn -keysize 512|1024|2048|4096|8192|
16384
```

- 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 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|4096|8192|16384
 -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
```

To add trust flags while adding a certificate to a wallet:

```
orapki wallet add -wallet wallet_location
-[trusted_cert|user_cert|self_signed]
-cert cert_location -pwd password -trust_flags "flag(s)"
```

- The `-trust_flags` parameter causes the specified flags to be added to the certificate. See Adding a Certificate with Trust Flags to Wallet for usage details.

See Adding a Certificate Request to an Oracle Wallet for an example showing the DN syntax.

orapki wallet change_pwd

Use this command to change the password for an Oracle wallet.

The syntax is:

```
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.

orapki wallet create

Use this command to create an Oracle wallet, to set auto-login on for an Oracle wallet, and to enable trust flags for certificates.

The syntax is:

```
orapki wallet create -wallet wallet_location
[-with_trust_flags] [-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.
- The `-with_trust_flags` parameter enables the wallet to support trust flags.

orapki wallet enable_trust_flags

Use this command to convert a wallet to support trust flags.

The syntax is:

```
orapki wallet enable_trust_flags -wallet wallet_location -pwd password
```

orapki wallet assign_trust_flags

Use this command to assign trust flags to a certificate in a wallet.

The syntax is:

```
orapki wallet assign_trust_flags [-wallet [wallet_location]] [-pwd password] [-
trust_flags ""|"flags"]
[-dn ["value"]] [-issuer [issuer_dn]] [-serial_num [serial_num]]
```

- The `-wallet` parameter specifies the location of the wallet from which you want to assign trust flags to a certificate.
- The `-pwd` specifies the wallet password.
- The `-trust_flags` parameter specifies which trust flags to enable. The trust flags are `SERVER_AUTH`,`CLIENT_AUTH`, `VALID_PEER`, and `NULL`.
- The `-dn` parameter specifies the distinguished name of the certificate.
- The `-issuer` option specifies the DN of the certificate issuer.
- The `-serial_num` option is used as an identification number for a certificate. The serial number option supports both decimal and hexadecimal format.

The `-serial_num` and `-issuer` options may be required to uniquely match a single certificate in the wallet.

For additional usage details, see Adding and Updating a Certificate's Trust Flags.

See Adding a Certificate Request to an Oracle Wallet for an example showing the DN syntax.

orapki wallet display

Use this command to view the certificate requests, user certificates, and trusted certificates in an Oracle wallet.

The syntax is:

```
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.

orapki wallet export

Use this command to export certificate requests and certificates from an Oracle wallet.

> **See Also:**
>
> Adding a Certificate Request to an Oracle Wallet for examples of specifying the `dn` parameter.

The syntax is:

```
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.

orapki wallet export_trust_chain

Use this command to export a chain of trust (certificate chain) for a user.

The syntax is:

```
orapki wallet export_trust_chain [-wallet [wallet]]
[-certchain [filename]]
[-dn [user_cert_dn] ]
[-pwd pwd]
[-issuer_dn [issuer_dn]]
[-serial_num [serial_num]]
```

- 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.

- The `-issuer_dn` option specifies the DN of the certificate issuer.
- The `-serial_num` option is used as an identification number for a certificate. The serial number option supports both decimal and hexadecimal format.

See Adding a Certificate Request to an Oracle Wallet for an example of how to specify the `–dn` parameter.

orapki wallet import_pkcs12

Use this command to import a PKCS#12 file into an Oracle wallet.

```
orapki wallet import_pkcs12
–wallet wallet_location [-pwd wallet_password]
-pkcs12file pkcs12_file_location [-pkcs12pwd pkcs12_file_password]
```

- The *`wallet`* parameter specifies the relative or absolute path of Oracle Wallet into which PKCS#12 file is to be imported. Required.
- The *`pwd`* parameter specifies the password of Oracle Wallet into which PKCS#12 file is to be imported. Optional, prompts as needed.
- The *`pkcs12file`* parameter specifies the relative or absolute path of PKCS#12 file to be imported into Oracle Wallet. Required.
- The *`pkcs12pwd`* parameter specifies the password of PKCS#12 file that is to be imported into Oracle Wallet. Optional, prompts as needed.

For example:

```
orapki wallet import_pkcs12 –wallet /scratch/user/oracleWalletFolder/ewallet.p12 -pwd
walletPassword -pkcs12file /scratch/userId/pkcs12fileFolder/certandkey.p12 -pkcs12pwd
pkcs12filePassword
```

H

Troubleshooting Oracle Fusion Middleware

You may encounter problems when using Oracle Fusion Middleware and need information on how to troubleshoot those problems.

- Diagnosing Oracle Fusion Middleware Problems
  To help diagnose Oracle Fusion Middleware problems, you can use its log files and the Diagnostic Framework for critical errors.
- Troubleshooting Common Problems and Solutions
  There are some common problems and solutions for Oracle Fusion Middleware.
- Need More Help?
  You can find more solutions on My Oracle Support or use can use the Remote Diagnostic Agent.

Diagnosing Oracle Fusion Middleware Problems

To help diagnose Oracle Fusion Middleware problems, you can use its log files and the Diagnostic Framework for critical errors.

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 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 Diagnosing Problems for more information about the Diagnostic Framework.

You can view an aggregated list of problems using the Support Workbench page of Fusion Middleware Control:

1. From the WebLogic Domain menu, select **Diagnostics,** then **Support** Workbench.
2. The summary page aggregates the Support Workbench diagnostic information across all WebLogic Servers that are members of this WebLogic Domain and shows the number of problems for each server. Click the number in the Problems column to see information about the problems for a particular server.

Troubleshooting Common Problems and Solutions

There are some common problems and solutions for Oracle Fusion Middleware.

This section describes some common problems and solutions. It contains the following topics:

Running Out of Data Source Connections

If the database performance has slowed or you receive the following message in the Oracle WebLogic Server log files, you may have leaks in the data source connections:

```
No resources currently available in pool datasource name

Any product functionality that depend on the datasource will not function as it can't
connect database to get required data.
```

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 specify explicitly 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 *Developing Applications for Oracle WebLogic Server*.

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 Remote Console or Fusion Middleware Control), you may receive a ClassNotFound error if Node Manager has not been configured to use the start scripts when starting Managed Servers. See Configuring Node Manager to Start Managed Servers for information about resolving this problem.

Troubleshooting SSL

This section describes common problems and solutions when working with SSL configuration. It contains the following topics:

Components May Enable All Supported Ciphers

You should be aware that when no cipher is explicitly configured, some 12*c* (12.2.1.1) components enable all supported SSL ciphers including `DH_Anon` (Diffie-Hellman Anonymous) ciphers.

At this time, Oracle HTTP Server is the only component known to set ciphers like this.

Configure the components with the desired cipher(s) if `DH_Anon` is not wanted.

SSL Certificate Chain Required on Certain Browsers

When you configure SSL for Oracle HTTP Server, you may need to import the entire certificate chain (rootCA, Intermediate CAs and so on).

Certain browsers, for example Internet Explorer, require that the entire certificate chain be imported to the browsers for SSL handshake to work. If your certificate was issued by an intermediate CA, you will need to ensure that the complete chain of certificates is available on the browser or the handshake will fail. If an intermediate certificate in the chain expires, it must be renewed along with all the certificates in the chain ((such as the OHS server certificate).

Troubleshooting FIPS Configuration

For details about this topic, see Troubleshooting FIPS 140 Issues.

Need More Help?

You can find more solutions on My Oracle Support or use can use the Remote Diagnostic Agent.

You can find more solutions on My Oracle Support, `https://support.oracle.com/signin`. 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:

- Using Remote Diagnostic Agent

Using Remote Diagnostic Agent

RDA is a command-line data collector and diagnostic tool which captures the right data when requested in a single step leading to faster issue resolution.

RDA collects the following information:

- Memory, CPU and disk data
- OS patches and packages
- Environment settings
- Network configuration and statistics
- Install, version and patch information
- Product configuration and log files
- Metrics (for example Status Info, DMS Dump, MBean values)

For more information about how to locate, upgrade, and use RDA, see Generating a RDA Report.