Oracle® Application Server

Administrator's Guide

10*g* Release 3 (10.1.3.4)

E10403-01

July 2008

ORACLE®

Oracle Application Server Administrator's Guide, 10g Release 3 (10.1.3.4)

E10403-01

Contributing Authors: Helen Grembowicz, Kevin Hwang, Peter LaQuerre, Mary Beth Roeser, Joe Ruzzi

Contributors: Steven Button, Pavana Jain, Michael Lehmann, Rose Pan, Thomas Van Raalte

Contents

3   Starting and Stopping

Part II Basic Administration

4 Managing Ports

5 Managing Log Files

Part III Advanced Administration

6 Reconfiguring Application Server Instances

7 Changing Network Configurations

8 Changing Infrastructure Services

9 Cloning Application Server Middle-Tier Instances

Part IV Secure Sockets Layer (SSL)

10 Overview of Secure Sockets Layer (SSL) in Oracle Application Server

11 Managing Wallets and Certificates

12   Enabling SSL in the Infrastructure

13 Enabling SSL in the Middle Tier

14 Troubleshooting SSL

Part V   Backup and Recovery

15 Introduction to Backup and Recovery

16 Oracle Application Server Recovery Manager

17 Backup Strategy and Procedures

18 Recovery Strategies and Procedures

19 Troubleshooting OracleAS Recovery Manager

Part VI   Appendixes and Glossary

A   Managing and Configuring Application Server Control

B   Oracle Application Server Command-Line Tools

C   URLs for Components

D   Oracle Application Server Port Numbers

E   Examples of Administrative Changes

F   Supplementary Procedures for Configuring LDAP-Based Replicas

G   Viewing Oracle Application Server Release Numbers

H   Troubleshooting Oracle Application Server

Glossary

Index

List of Figures

List of Tables

Preface

This guide describes how to manage Oracle Application Server, including how to start and stop Oracle Application Server, how to reconfigure components, and how to back up and recover Oracle Application Server.

Audience

This guide is intended for administrators of Oracle Application Server.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documentation

For more information, see these Oracle resources:

- Oracle Application Server Documentation Library

- Oracle Application Server Platform-Specific Documentation

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN):

http://www.oracle.com/technology/documentation/

Conventions

The following text conventions are used in this document:

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

What's New in Oracle Application Server Administration?

This preface introduces the new and changed administrative features of Oracle Application Server 10*g* Release 3 (10.1.3.4). This information is mostly useful to users who have managed previous releases of Oracle Application Server, including Oracle Application Server 10*g* Release 2 (10.1.2), or 10g Release 3 (10.1.3.*).

New Features for 10*g* Release 3 (10.1.3.4)

The new administrative features of Oracle Application Server 10*g* Release 3 (10.1.3.4) include:

- A single Oracle Application Server installation (`ORACLE_HOME`) can be used to create multiple application server instances. The instances can be local to `ORACLE_HOME`, in a different directory on the same host as `ORACLE_HOME`, or distributed across hosts on the network. For more information, see Section 6.1.

Part I

Getting Started

This part contains information for getting started with managing Oracle Application Server.

It contains the following chapters:

- Chapter 1, "Getting Started After Installing Oracle Application Server"
- Chapter 2, "Introduction to Administration Tools"
- Chapter 3, "Starting and Stopping"

1

Getting Started After Installing Oracle Application Server

This chapter contains information and tasks to help you get started managing Oracle Application Server after installation.

It contains the following topics:

- Understanding Oracle Application Server 10g Release 3 (10.1.3.4)

- Task 1: Set Up Environment Variables

- Task 2: Use the Oracle Application Server Welcome Page

- Task 3: Check Your Port Numbers

- Task 4: Get Started with Managing Components

- Task 5: Enable SSL (Optional)

1.1 Understanding Oracle Application Server 10*g* Release 3 (10.1.3.4)

Oracle Application Server 10*g* Release 3 (10.1.3.4) provides a complete Java 2, Enterprise Edition (J2EE) 1.4-compliant environment.

Depending on the installation type, it includes Oracle HTTP Server, Oracle Containers for J2EE (OC4J), Oracle Process Manager and Notification Server (OPMN), Application Server Control, OC4J Java Single Sign-On, Oracle WebCenter Framework, Oracle Content DB, and Oracle Business Rules.

You can use this release with Release 10.1.4 or Release 2 (10.1.2) Oracle Identity Management Services and Release 2 (10.1.2) Oracle Application Server Web Cache.

1.2 Task 1: Set Up Environment Variables

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

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

- Table 1–1, " Environment Variables for UNIX"

- Table 1–2, " Environment Variables for Windows"

Table 1–1   Environment Variables for UNIX

Environment Variable	Value
DISPLAY	*hostname*:*display_number*.*screen_number*
	Beginning with Oracle Application Server 10*g*, very few tools, such as `oidadmin`, require the DISPLAY variable.
LD_LIBRARY_PATH	On Solaris, make sure the value contains the following directory:
	`$ORACLE_HOME/lib32`
	On Linux, make sure the value contains the following directory:
	`$ORACLE_HOME/lib`
	On HP-UX, make sure the value contains the following directory:
	`$ORACLE_HOME/lib`
	On IBM AIX, make sure this environment variable is not set.
(IBM AIX only) LIBPATH	If the calling application is a 32-bit application, make sure the value contains the following directory:
	`$ORACLE_HOME/lib32`
	If the calling application is a 64-bit application, make sure the value contains the following directory:
	`$ORACLE_HOME/lib`
(Solaris only) LD_LIBRARY_PATH_64	Make sure the value contains the following directory:
	`$ORACLE_HOME/lib`
(HP-UX only) SHLIB_PATH	Make sure the value contains the following directory:
	`$ORACLE_HOME/lib32`
ORACLE_HOME	Set to the full path of the installation's Oracle home. (When you run root.sh as the root user during the installation, this variable is set.)
PATH	Make sure the value contains the following directories, which contain basic commands used by all installations:
	`$ORACLE_HOME/bin` `$ORACLE_HOME/opmn/bin`
	When you start to work with specific components, you may want to add additional directories to your path, as recommended by the component documentation.

Table 1–2 shows the environment variables for Windows.

Table 1–2   Environment Variables for Windows

Environment Variable	Value
ORACLE_HOME	Set to the full path of the installation's Oracle home.
	The value is automatically set by Oracle Universal Installer.
TEMP	Set to your temp directory, for example, `C:\temp`.
TMP	Set to your temp directory, for example, `C:\temp`.

Best Practices for Multiple Installations on a UNIX Host

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

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

1. Log in as the user who installed the installation you want to work on.

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

   ```
   su - user
   ```

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

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

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

1.3 Task 2: Use the Oracle Application Server Welcome Page

The Oracle Application Server Welcome Page is a great starting point for managing your application server. It includes the following:

- An overview of this release

- A link to the Oracle Enterprise Manager 10*g* Application Server Control Console—a Web-based tool for managing Oracle Application Server

- Links to additional resources

Figure 1–1 shows part of the Welcome Page.

Figure 1–1   Oracle Application Server Welcome Page

Accessing the Welcome Page

You can locate the URL for accessing the Welcome Page on the End of Installation Screen text, which is stored in the following file:

```
(UNIX) ORACLE_HOME/install/readme.txt
(Windows) ORACLE_HOME\install\readme.txt
```

To view the Welcome Page, connect to it using the HTTP listener port on your installation. For example:

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

If you select the Basic installation option, the default port is 7777 on UNIX and 80 on Windows.

> **Tip**   If you cannot access the Welcome Page, try the following:

1. Check `readme.txt` and make sure you are using the correct URL (hostname and port number).

2. Try restarting Oracle HTTP Server:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=HTTP_Server
(UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=HTTP_Server

(Windows) ORACLE_HOME\opmn\bin\opmnctl stopproc ias-component=HTTP_Server
(Windows) ORACLE_HOME\opmn\bin\opmnctl startproc ias-component=HTTP_Server
```

1.4 Task 3: Check Your Port Numbers

During installation, Oracle Application Server assigned port numbers to various components and services. It is important to check these port numbers because:

- You need to know these port numbers in order to start managing your application server.

- Oracle Application Server takes several measures to ensure that port number assignments are unique. However, it is possible that a port assignment could conflict with a non-Oracle Application Server process on your host that was not running during the installation. If you determine there is a conflict, stop the non-Oracle Application Server process and continue with the tasks in this chapter. Once you have completed the tasks in this chapter and have verified that your installation is running properly, you can consider changing Oracle Application Server port numbers.

> **See Also:**   Chapter 4 for information on changing port numbers

You can find the list of port numbers used by typing the following command:

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

The following shows the output from an installation on Linux (some columns are not shown to improve readability):

```
Processes in Instance: orcl10132.myhost.ora.com
------------------------+--------------------+--------+-----+----------------------------------------
ias-component           | process-type       | pid-   | ... | ports
------------------------+--------------------+--------+-----+----------------------------------------
OC4JGroup:default_group | OC4J:OC4J_Content  | 27769  | ... | jms:12602,ajp:12502,rmis:12702,rmi:12402
OC4JGroup:default_group | OC4J:OC4J_WebCent~ | 27501  | ... | jms:12601,ajp:12501,rmis:12701,rmi:12401
```

```
OC4JGroup:default_group  | OC4J:home           | 27394  | ... | jms:12603,ajp:12503,rmis:12703,rmi:12403
HTTP_Server              | HTTP_Server         | 26928  | ... |https1:4443,http2:7200,http1:7777
ASG                      | ASG                 | N/A    | ... | N/A
```

You should leave the port numbers as they are until you have completed the tasks in this chapter and confirmed that all of your components are running properly. Then, you can consider changing port numbers. Note that some port numbers cannot be changed, and some require additional steps for updating other components.

On Windows, you can view the port numbers from the Windows Start menu. For example on Windows 2000, select **Start** > **Programs** > **Oracle** - *Oracle_home_name* > **Oracle Process Manager** > **Oracle Assigned Port Numbers**.

1.5 Task 4: Get Started with Managing Components

This section describes how to begin managing components. It includes instructions for accessing component administration tools and pointers to more information. It contains the following topics:

- Getting Started with Oracle Process Manager and Notification Server (OPMN)

- Getting Started with Oracle HTTP Server

- Getting Started with Oracle Containers for J2EE (OC4J)

- Getting Started with Oracle WebCenter Framework

- Getting Started with Oracle Content DB

> **See Also:**   Appendix C for a quick reference on how to access components

1.5.1 Getting Started with Oracle Process Manager and Notification Server (OPMN)

Oracle Process Manager and Notification Server (OPMN) manages and monitors most Oracle Application Server components. It is installed and configured in every middle-tier installation and is essential for running Oracle Application Server.

OPMN provides the opmnctl command. The executable file is located in the following directory:

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

To get started with OPMN, use the opmnctl command to query the status of the components in your installation:

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

Example 1–1 shows sample output from the command. It displays the component name, process type, operating system process ID (PID), and status of each process.

Example 1–1   Sample Output from opmnctl status Command

```
Processes in Instance: orcl10132.myhost.ora.com
-----------------------+-------------------+---------+---------
ias-component          | process-type      |    pid  | status
-----------------------+-------------------+---------+---------
OC4JGroup:default_group | OC4J:OC4J_Content  |   27769 | Alive
OC4JGroup:default_group | OC4J:OC4J_WebCent~ |   27501 | Alive
OC4JGroup:default_group | OC4J:home          |   27394 | Alive
```

```
HTTP_Server              | HTTP_Server            |   26928 | Alive
ASG                      | ASG                    |     N/A | Down
```

You can use OPMN to start and stop your application server, monitor components, configure event scripts, and perform many other tasks related to process management. For example, you can use the following commands on UNIX to start and stop OPMN and all OPMN-managed processes, such as Oracle HTTP Server and OC4J instances:

```
ORACLE_HOME/opmn/bin/opmnctl startall
ORACLE_HOME/opmn/bin/opmnctl stopall
```

On Windows, you can invoke these commands from the Windows Start menu. For example to start all processes, on Windows 2000, select **Start** > **Programs** > **Oracle** - *Oracle_home_name* > **Oracle Process Manager** > **Start Oracle Process Manager**.

> **See Also:** *Oracle Process Manager and Notification Server Administrator's Guide*

Note that the ASG process, which is the OracleAS Guard server, is not started after you finish installing Oracle Application Server. If you are using OracleAS Guard, you can start it after installation.

> **See:** *Oracle Application Server High Availability Guide*

1.5.2 Getting Started with Oracle HTTP Server

Oracle HTTP Server is installed and configured when you install the following installation types:

- Basic Installation
- Advanced Installation: Oracle WebCenter Framework with Oracle HTTP Server
- Advanced Installation: Oracle Content Database
- Advanced Installation: Oracle HTTP Server

You can access Oracle HTTP Server as follows:

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

In the example, *port* is the Oracle HTTP Server Listen port number, which is listed in the output from the opmnctl status -l command.

The following example assumes that the port is 7777:

```
http://hostname.domain:7777
```

When you access Oracle HTTP Server, you see the Oracle Application Server Welcome Page. Click **Application Server Control** to log in to Application Server Control. Then, navigate to the Application Server:*server_name* Home page to see the status of the HTTP server.

> **See Also:**
>
> - *Oracle HTTP Server Administrator's Guide*
> - Section 2.3, "Getting Started with the Application Server Control"

1.5.3 Getting Started with Oracle Containers for J2EE (OC4J)

Oracle Containers for J2EE (OC4J) is a complete Java 2 Enterprise Edition (J2EE) environment.

When you install Oracle Application Server, the `home` OC4J instance is created. This is the default OC4J instance that comes with every middle-tier installation. If you install Oracle WebCenter Framework, an OC4J instance, named `OC4J_WebCenter`, is created. If you install Oracle Content DB, another OC4J instance, named `OC4J_Content`, is created.

You can use Application Server Control to manage OC4J instances by navigating to the Home page for the instance.

> **See Also:**
>
> - *Oracle Containers for J2EE Configuration and Administration Guide*
> - Section 2.3.5, "Using Application Server Control to View and Manage an OC4J Instance"

1.5.4 Getting Started with Oracle Business Rules

Oracle Business Rules enables applications to rapidly adapt to regulatory and competitive pressures. Business rules are statements that describe business policies. Oracle Business Rules Rule Author lets you work with rules from anywhere using a web browser and provides a point-and-click interface for creating new rules and editing existing rules.

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

In the example, *port* is the Oracle HTTP Server Listen port number, which is listed in the output from the `opmnctl status -l` command.

> **See Also:** *Oracle Business Rules User's Guide*

1.5.5 Getting Started with Oracle TopLink

You can choose to install Oracle TopLink if you select the Advanced installation option. If you have installed Oracle TopLink, see the *Oracle TopLink Getting Started Guide* for information about getting started with Oracle TopLink.

1.5.6 Getting Started with Oracle Enterprise Service Bus

Oracle Enterprise Service Bus is the foundation for services using service-oriented architecture (SOA) and event-driven architecture (EDA). At its core, it is a loosely coupled application framework that provides your business with increased flexibility, reusability, and overall responsiveness in a distributed, heterogeneous, message-oriented environment using industry standards.

To access the Oracle Enterprise Service Bus console, use the following URL:

`http://host:port/esb`

In the example, *port* is the Oracle HTTP Server Listen port number, which is listed in the output from the `opmnctl status -l` command.

You can also access the console from the Welcome page.

Alternatively, on Windows: From the desktop **Start** button, select **All Programs > Oracle – *Oracle_Home* > Oracle ESB 10.1.3.1.0 > ESB Control**, where *Oracle_Home* is the name of the Oracle home where you installed Oracle Enterprise Service Bus.

> **See Also:** *Oracle Enterprise Service Bus Developer's Guide*

1.5.7 Getting Started with Oracle BPEL Process Manager

With Oracle BPEL Process Manager, you can design service-oriented architecture (SOA)-based applications by developing synchronous and asynchronous services into end-to-end BPEL process flows.

You can access the Oracle BPEL Process Manager console using the following URL:

```
http://host:port/BPELConsole
```

In the example, *port* is the Oracle HTTP Server Listen port number, which is listed in the output from the opmnctl status -l command. You can also access the console from the Welcome page.

Alternatively, on Windows: From the desktop **Start** button, select **All Programs > Oracle – *Oracle_Home* > Oracle BPEL Process Manager 10.1.3.1.0 > BPEL Console**, where *Oracle_Home* is the name of the Oracle home where you installed Oracle BPEL Process Manager.

> **See Also:** *Oracle BPEL Process Manager Administrator's Guide*

1.5.8 Getting Started with Oracle Web Services Manager

Oracle Web Services Manager allows companies to define policies that govern Web service operations such as access, authorization, logging, and load balancing, and then wrap these policies around Web services.

You can access the Oracle Web Services Manager Control using the following URL:

```
http://host:port/ccore
```

In the example, *port* is the Oracle HTTP Server Listen port number, which is listed in the output from the opmnctl status -l command.

You can also access Oracle Web Services Manager Control from the Welcome page.

> **See Also:** *Oracle Web Services Manager User and Administrator Guide*

1.5.9 Getting Started with Oracle WebCenter Framework

Oracle WebCenter Framework is a part of Oracle WebCenter Suite. Oracle WebCenter Suite is a set of tools and services that you can use to quickly and easily build transactional, analytical, and dynamic applications. You get portlets, content, run-time customization, presence, collaboration, and search—everything you need to maximize the potential of your service-oriented applications.

Oracle WebCenter Framework enables you to build portal features directly into your Java EE 5 applications and bring traditional portal capabilities, like portlet and page customization, to your users. Oracle WebCenter Framework is the first offering that enables you to interweave those portal capabilities directly into a JavaServer Faces page using standard Faces constructs.

> **See:** *Oracle WebCenter Framework Tutorial*

1.5.10 Getting Started with Oracle Content DB

Oracle Content DB is a consolidated, database-centric content management application that provides a comprehensive, integrated solution for file and document lifecycle management.

You can use Application Server Control to manage Oracle Content DB. Navigate to Application Server Control Console from the Welcome page. From the Cluster Topology page, expand the OC4J_Content instance, and click **content.** Then, click **Content DB Extension.**

> **See Also:** *Oracle Content Database for Oracle WebCenter Suite Administrator's Guide*

1.6 Task 5: Enable SSL (Optional)

During installation, SSL is not configured for some components. If you would like to enable SSL, refer to Part IV, "Secure Sockets Layer (SSL)".

2

Introduction to Administration Tools

This chapter introduces the Oracle Application Server administration tools.

It contains the following topics:

- Overview of Oracle Application Server Administration Tools
- About Oracle Enterprise Manager 10g Application Server Control
- Getting Started with the Application Server Control

2.1 Overview of Oracle Application Server Administration Tools

Oracle realizes that the procedures you use to monitor and administer your application server components can vary, depending upon the size of your organization, the number of administrators you employ, and the types of components you manage. As a result, Oracle offers options for managing your Oracle Application Server installations.

These management options can be divided into the following categories:

- Managing Oracle Application Server with Oracle Enterprise Manager 10g Application Server Control
- Managing Oracle Application Server Using the OPMN Command Line
- Managing Oracle Application Server Using the admin_client.jar Utility
- Using Other Tools to Monitor the Built-In Performance Metrics

2.1.1 Managing Oracle Application Server with Oracle Enterprise Manager 10*g* Application Server Control

Oracle Enterprise Manager 10*g* Application Server Control is installed with every instance of Oracle Application Server. As a result, you can immediately begin managing your application server and its components from your Web browser.

From Application Server Control, you can monitor and administer a single Oracle Application Server instance or a **cluster topology,** which consists of multiple application server instances and the applications deployed to the OC4J instances within those application servers.

Application Server Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for specific features of the application server and Oracle Containers for J2EE. The Enterprise Manager home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions—all from your Web browser.

> **See Also:** Section 2.2, "About Oracle Enterprise Manager 10g Application Server Control"

2.1.2 Managing Oracle Application Server Using the OPMN Command Line

In addition to Application Server Control, you can also use the `opmnctl` command-line tool, which provides a command-line interface to Oracle Process Manager and Notification Server (OPMN). For example, you can use this command-line tool (`opmnctl`) to:

- Start and stop applications and application server instances
- Associate two or more application server instances to form a cluster
- Display a summary of the instances that are part of a cluster

> **See Also:**
>
> - Section 1.5.1, "Getting Started with Oracle Process Manager and Notification Server (OPMN)"
> - *Oracle Containers for J2EE Configuration and Administration Guide* for information about configuring and managing OC4J clusters with `opmnctl`
> - *Oracle Process Manager and Notification Server Administrator's Guide* for complete information about the OPMN command line

2.1.3 Managing Oracle Application Server Using the admin_client.jar Utility

Oracle Application Server also provides a command-line utility, `admin_client.jar`, that can be used to perform operations on active OC4J instances.

For many functions, the `admin_client.jar` utility replaces the `admin.jar` utility, which is used exclusively for the standalone configuration of the Oracle Application Server OC4J server.

Unlike the `admin.jar` utility, you can use the `admin_client.jar` utility to manage OC4J instances in a managed Oracle Application Server environment, as well as OC4J instances in a standalone OC4J environment.

You can perform the following tasks with the `admin_client.jar` utility:

- Deploy applications to a specific OC4J instance or to all instances within a cluster
- Undeploy an application
- Incrementally update a deployed EJB module with modified classes
- Create a new shared library
- Stop, start, or restart a specific application, on a specific OC4J instance or cluster-wide

> **See Also:** "Using the admin_client.jar Utility" in the *Oracle Containers for J2EE Configuration and Administration Guide*

2.1.4 Using Other Tools to Monitor the Built-In Performance Metrics

After you install and start Oracle Application Server, the application server automatically begins gathering a set of built-in performance metrics. These built-in performance metrics are measured continuously using performance instrumentation inserted into the implementations of Oracle Application Server components.

Application Server Control presents a subset of these performance metrics in an organized fashion on the application server component home pages. For example, the OC4J performance metrics are presented as a series of charts on the OC4J Performance page.

Alternatively, you may want to view the complete set of built-in performance metrics, or you may need to monitor a specific set of application server component metrics. Oracle Application Server provides a set of command-line and servlet-based tools to view the Oracle Application Server built-in performance metrics directly, outside of Application Server Control.

> **See Also:** *Oracle Application Server Performance Guide*

2.2 About Oracle Enterprise Manager 10*g* Application Server Control

Oracle Enterprise Manager 10*g* Application Server Control provides Web-based management capabilities designed specifically for Oracle Application Server 10*g* Release 3 (10.1.3.*x*).

The following sections provide more information:

- Application Server Control New Features for 10g Release 3 (10.1.3.0.0)
- New Application Server Control Features Introduced for 10g Release 3 (10.1.3.1)
- About the Application Server Control Underlying Technologies
- Using the Application Server Control Online Help

2.2.1 Application Server Control New Features for 10*g* Release 3 (10.1.3.0.0)

For Oracle Application Server 10*g* Release 3 (10.1.3.0.0), Application Server Control introduced the following benefits and feature improvements. These features are also part of the 10*g* Release 3 (10.1.3.1), 10*g* Release 3 (10.1.3.2), and 10*g* Release 3 (10.1.3.4) releases:

- Lightweight Architecture
- Standards-Based Management
- Remote Management
- Role-Based Administration

> **Note:** In the following sections, 10*g* Release 3 (10.1.3.*x*) refers to 10*g* Release 3 (10.1.3.0.0), 10*g* Release 3 (10.1.3.1), 10*g* Release 3 (10.1.3.2), 10*g* Release 3 (10.1.3.3) and 10*g* Release 3 (10.1.3.4).

2.2.1.1 Lightweight Architecture

The Application Server Control for 10*g* Release 3 (10.1.3.*x*) is deployed as a standard J2EE application (`ascontrol`) that runs within every OC4J container you create. In small deployments, the OC4J instance that hosts the Application Server Control can also be used to deploy customer applications.

This new architecture eliminates the need for a separate Oracle Management Agent, which was required for the Application Server Control in previous Oracle Application Server releases. As a result, the management software requires less disk space and fewer system resources. You can then manage and monitor your applications with minimal impact on your application server environment.

2.2.1.2 Standards-Based Management

For 10*g* Release 3 (10.1.3.*x*), the Application Server Control is based on the Java Management Extensions (JMX) technology.

> **See Also:** The Java Management Extensions (JMX) Technology Overview at:
>
> `http://java.sun.com/j2se/1.5.0/docs/guide/jmx/overview/JMXoverviewT OC.html`

Specifically, Application Server Control implements the following Java technology standards:

- The JSR 77 J2EE Management specification

  Application Server Control takes advantage of container-defined MBeans provided by the 10*g* Release 3 (10.1.3.*x*) OC4J container, as well as application-defined MBeans, provided by applications that are developed and configured in accordance with the specification. These MBeans provide configuration, monitoring, and state management capabilities.

  > **See Also:** The JSR 77 J2EE Management Specification at:
  >
  > `http://jcp.org/en/jsr/detail?id=077`

  In addition, Application Server Control includes a new MBean browser that provides complete support for JSR 77 and Application Defined MBean operations, such as viewing MBean attribute values and statistics, MBean method invocation, subscription to JMX notifications, and state management.

  > **See Also:** Section 2.3.6, "About MBeans and the Application Server Control MBean Browsers"

- The JSR 88 J2EE Application Deployment specification

  Application Server Control provides a JSR 88-based deployment wizard to simplify deployment and redeployment of J2EE applications, task-oriented deployment plan editors to assist you in assigning or mapping the common deployment descriptors at deploy time, and a generic deployment plan editor so you can access all deployment descriptors for advanced configuration.

  > **See Also:** The JSR 88 J2EE Application Deployment specification at:
  >
  > `http://www.jcp.org/en/jsr/detail?id=88`

- The Java Naming and Directory Interface (JNDI)

  Application Server Control provides a new JNDI browser, which hierarchically displays the JNDI bindings for each application.

  > **See Also:** Description of the Java Naming and Directory Interface (JNDI) at:
  >
  > `http://java.sun.com/products/jndi/`

- Web Services Management

  From the Application Server Control, you can configure auditing, logging, security, and reliability parameters for Web Services deployed in your OC4J instances. In addition, you can use Oracle Web Services Manager as your exclusive

Web Services management solution, or you can use it in concert with the standard auditing, logging, reliability, and security management features.

> **See Also:** *Oracle Application Server Web Services Developer's Guide* for information about developing and managing Web services.
>
> *Oracle Web Services Manager User and Administrator Guide* for more information about Oracle Web Services Manager

2.2.1.3 Remote Management

With Oracle Application Server 10*g* Release 3 (10.1.3.*x*), you can use Oracle Process Manager and Notification Server (OPMN) to associate two or more application server instances to form an Oracle Application Server cluster. When you configure your environment in this manner, you can use a single instance of the Application Server Control to remotely manage all the instances in the cluster.

> **See Also:** Section 2.3.3.1, "Viewing the Cluster Topology and Locating the Active Application Server Control"

2.2.1.4 Role-Based Administration

In a typical production data center, administrative operations (for example, configuration, application deployment, and process control and monitoring) are performed by different sets of administrators, depending upon the data center deployment and security policies.

Typically a super administrator, who is granted full privileges, delegates administrative responsibilities to other administrators who are assigned limited privileges to carry out only specific administrative operations. With Application Server Control, you can assign one of three standard administrative roles to each user.

> **See Also:** Section 2.3.2, "Creating Administration Accounts for You and Your Team"

2.2.2 New Application Server Control Features Introduced for 10*g* Release 3 (10.1.3.1)

For 10*g* Release 3 (10.1.3.1), Application Server Control offered the following new features. These features are also available in 10*g* Release 3 (10.1.3.2), 10*g* Release 3 (10.1.3.3), and 10*g* Release 3 (10.1.3.4).

- The ability to create and delete OC4J instances from Application Server Control

- The ability to create and delete groups from Application Server Control

- A new Runtime Ports page that summarizes all the ports currently in use by components of the Oracle Application Server cluster topology

- A new Routing ID page that allows you to view and modify the routing ID assigned to your application servers, OC4J instances, and deployed applications

- A new Topology Network Configuration page that you can use to configure the members and the properties of the cluster topology

For more information, see Section 2.3.

2.2.3 About the Application Server Control Underlying Technologies

Application Server Control relies on related technologies to discover, monitor, and administer the Oracle Application Server environment. These technologies automatically perform many of the management tasks as you select options and

functions within Application Server Control. For example, they discover the components of each application server instance, gather and process performance data, and provide access to application configuration information.

Table 2–1 provides a summary of the underlying technologies leveraged by Application Server Control.

Table 2–1   Summary of the Application Server Control Underlying Technologies

Technology	Description
Dynamic Monitoring Service (DMS)	Application Server Control uses DMS to gather performance data about your Oracle Application Server components.
	For more information, see *Oracle Application Server Performance Guide*.
Oracle Process Manager and Notification Server (OPMN)	OPMN provides process control and monitoring for application server instances and their components. It gathers component status information, and distributes the status information to components that are interested in it. Application Server Control uses OPMN for such tasks as starting and stopping the components of your application server instance.
	For more information, see *Oracle Process Manager and Notification Server Administrator's Guide*.

2.2.4 Using the Application Server Control Online Help

At any time while using the Application Server Control, you can click **Help** at the top of the page to get more information. In most cases, the Help window displays a help topic about the current page. Click **Contents** in the Help window to browse the list of help topics, or click **Search** to search for a particular word or phrase.

2.3 Getting Started with the Application Server Control

Use the following sections to get started with the Application Server Control and become familiar with the Enterprise Manager home pages within the Application Server Control:

- Displaying the Application Server Control

- Creating Administration Accounts for You and Your Team

- Using Application Server Control to Manage the Cluster Topology

- Using Application Server Control to Manage the Application Server Components

- Using Application Server Control to View and Manage an OC4J Instance

- About MBeans and the Application Server Control MBean Browsers

2.3.1 Displaying the Application Server Control

The following sections describe how to display Application Server Control and introduce you to the initial home pages you should see when you display Application Server Control for the first time:

- Using the Application Server Control URL

- Displaying Application Server Control from the Welcome Page

2.3.1.1 Using the Application Server Control URL

The URL for Application Server Control, including the port number, is included in the text file that displays at the end of the Oracle Application Server installation

procedure. This text file is saved in the following location after you install the application server:

```
(UNIX) ORACLE_HOME/install/readme.txt
(Windows) ORACLE_HOME\install\readme.txt
```

The Application Server Control URL typically includes the name of the host computer and the port number assigned to Application Server Control during the installation. For example, on UNIX:

```
http://mgmthost1.acme.com:7777/em
```

2.3.1.2 Displaying Application Server Control from the Welcome Page

To view Application Server Control from the Oracle Application Server Welcome Page:

1. Display the Oracle Application Server Welcome Page by entering the following URL in your Web browser:

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

   For example:

   ```
   http://sys42.acme.com:7777
   ```

   > **Note:** The default port for Oracle HTTP Server (and, as a result, the Welcome page) is provided at the end of the Oracle Application Server installation, as well as in the following text file in the `install` directory of the application server Oracle home:
   >
   > ```
   > readme.txt
   > ```

2. Click **Log on to the Oracle Enterprise Manager 10*g* Application Server Control Console**.

   Enterprise Manager displays the administrator logon dialog box.

3. Enter the Oracle Application Server administrator user name and password and click **OK**.

   The default user name for the administrator user is `oc4jadmin`. This is the account you can use to log in to Application Server Control for the first time. The `oc4jadmin` password is the one you supplied during the installation of Oracle Application Server.

2.3.2 Creating Administration Accounts for You and Your Team

After you log in to Application Server Control by using the default `oc4jadmin` account, use the following procedure to create a new administration user account for yourself and additional administrative user accounts for the system administrators on your team.

Oracle recommends that you do not use the `oc4jadmin` account for everyday administration tasks. Instead, the `oc4jadmin` account should be reserved and used exclusively as the administration credentials for the cluster. For more information, see Section A.2.2.

Task 1 Create a New Administrator Account for Yourself

1. Click **Setup** at the top of any Application Server Control page.

2. Click **Users** to display the Users page.

3. Click **Create** to create a new administrator account.

4. Follow the instructions on the screen to provide a name for your account, a default password, and to assign the `ascontrol_admin` role to your administrator account.

   Consider using a combination of your last name and your first name initial as a name for the account (for example, `bsmith`).

   It is important that you assign yourself the ascontrol_admin role so you can perform all administration tasks and create additional administration users.

Task 2 Create Additional Administration Accounts for Members of Your Team

Follow the procedure in the previous step to create additional user accounts for the members of your team.

Refer to Table 2–2 for a description of the administrative roles you can assign to each user.

Table 2–2   Administrative Roles for Application Server Control Administrators

Role	Description
ascontrol_admin	Assign this role to administrators who need complete administrative privileges and access to Application Server Control application and pages. You can consider these administrators the "super users" assigned to manage your Oracle Application Server and OC4J environment.
ascontrol_appadmin	Assign this role to administrators who need to manage all deployed applications, except the default and Application Server Control (ascontrol) applications. These administrators can manage the applications you deploy, but they cannot create new administrative users or make global configuration changes.
ascontrol_monitor	Assign this role to administrators who must monitor the Oracle Application Server and OC4J environment, but do not need to make any configuration changes to the applications or the OC4J instance. This role is essentially a read-only role.
	This role is applied automatically to all administrative users that you create. If you do not apply the ascontrol_admin or ascontrol_appadmin role, then the account can be used for monitoring only.

2.3.3 Using Application Server Control to Manage the Cluster Topology

When you install Oracle Application Server and then log in to Application Server Control, the first page you see is the Cluster Topology page. The following sections describe how you can use this page to get started managing your Oracle Application Server environment:

- Viewing the Cluster Topology and Locating the Active Application Server Control

- Using Application Server Control to Manage Groups

- Summary of the Cluster Topology Administration Tasks

> **See Also:**
>
> - *Oracle Application Server Installation Guide* for information on configuring a cluster during the Oracle Application Server installation
> - Section 6.3, "Configuring Cluster Topologies" for information on configuring clusters after installation

2.3.3.1 Viewing the Cluster Topology and Locating the Active Application Server Control

When you install Oracle Application Server, the Cluster Topology page (Figure 2–1) is the first page you see when you log in to Application Server Control. You can also view this page at any time by clicking **Cluster Topology** in Application Server Control.

The Cluster Topology page provides you with a high-level view of the application servers, OC4J instances, Web servers, and applications deployed to the cluster.

Note that each OC4J instance in the cluster automatically includes an `ascontrol` application, which represents the Application Server Control. However, only one Application Server Control is used to manage all the Oracle Application Server instances in the cluster.

❖ To identify the active Application Server Control, which is being used to manage the cluster, click **Expand All** to view all components of the cluster, and then locate the active `ascontrol` application, which is identified by the active Application Server Control icon.

Note that the OC4J instance that is used to deploy the active `ascontrol` application is called the Administration OC4J instance. You can identify the Administration OC4J instance during the Oracle Application Server installation procedure.

> **See Also:** Section A.6, "Managing the Active Application Server Control" for more information about identifying and configuring the active Application Server Control for a cluster topology

The Cluster Topology page also displays any applications that you deploy. To view all the applications deployed to the OC4J instances in the cluster, select **Applications** from the **View By** menu above the Members section of the page. Application Server Control organizes the list into distinct categories.

Figure 2–1   Managing the Cluster Topology

2.3.3.2 Using Application Server Control to Manage Groups

In Oracle Application Server 10*g* Release 3 (10.1.3.*x*), a **group** is a set of OC4J instances that belong to the same cluster topology. Specific configuration operations can be executed simultaneously on all OC4J instances in the group.

For more information, refer to the following sections:

- Viewing and Managing Groups

- Advantages of Using Groups

2.3.3.2.1   Viewing and Managing Groups When you first install Oracle Application Server, a default group is created for you. The `default_group` contains any OC4J instances created during the installation.

After installation, the groups available in the cluster are listed in the Groups section of the Cluster Topology page (Figure 2–1). From the Groups section of the page, you can start, stop, delete, and create groups. You can also click the name of a group and use the resulting Group page. From the Group page, you can:

- Move OC4J instances in and out the group

- Deploy applications to the group

- Perform certain administration tasks on the group

Every OC4J instance must be in a group. As a result, when you create a new OC4J instance, you must identify the group to which it will belong.

See Also:

- Section 6.3.3, "Creating Additional Groups"
- "About Groups" in the Application Server Control online help

2.3.3.2.2 Advantages of Using Groups Groups allow you to perform some common administrative tasks automatically across multiple OC4J instances.

Specifically, from the Group page, you can, in one step, perform each of the following tasks across multiple OC4J instances:

- Process management operations, such as start, stop, or restart
- Deployment operations, such as deploy, undeploy, and redeploy
- JDBC management operations, such as creating, modifying, or removing JDBC data sources and connection pools
- JMS Provider operations, such as creating and removing JMS destinations, and creating, modifying, or removing JMS connection factories

To display the Group page, simply click the name of the group in the Groups section of the Cluster Topology page.

> **See Also:** "Guidelines for Creating OC4J Instances and Groups" in the Application Server Control online help

2.3.3.3 Summary of the Cluster Topology Administration Tasks

From the Cluster Topology Page, you can perform a set of cluster-wide administration tasks from the Administration section of the page. Table 2–3 summarizes the cluster topology administration tasks.

Table 2–3   Summary of the Cluster Topology Administration Tasks

Task	Description	More information
Cluster MBean Browser	Display the Cluster MBean Browser, which you can use to view the hierarchy of managed beans that are specific to cluster-wide operations.	Section 2.3.6, "About MBeans and the Application Server Control MBean Browsers"
Routing ID Configuration	Display or modify the routing IDs currently assigned to the components of your Oracle Application Server cluster.	"Changing Routing IDs" in the Application Server Control online help
Java SSO Configuration	Configure deployed applications to use OC4J Java Single Sign-On (Java SSO), a lighter-weight single sign-on solution supplied with OC4J that does not require additional infrastructure. Note that for a Basic Installation, Java SSO is automatically deployed, configured, and started. For an Advanced Installation, it is deployed, but not configured or started.	"Java SSO Setup and Configuration" in the *Oracle Containers for J2EE Security Guide*

Table 2–3 (Cont.) Summary of the Cluster Topology Administration Tasks

Task	Description	More information
Topology Network Configuration	Configure the members and properties of the current Oracle Application Server cluster topology.	Section 6.3, "Configuring Cluster Topologies" "Summary of the Supported Cluster Topologies" in the Application Server Control online help
Runtime Ports	View and modify the ports used by the components of your Oracle Application Server cluster.	Chapter 4, "Managing Ports"

2.3.4 Using Application Server Control to Manage the Application Server Components

After you get familiar with the Cluster Topology page, you can then drill down to the Home page for a particular application server instance.

Click the name of the application server instance on the Cluster Topology to display the Application Server page.

The resulting Application Server page is shown in Figure 2–2.

Figure 2–2 Viewing the Components of the Application Server Instance

This page lists the components of the application server instance, including the OC4J instances created in this instance and, if one was installed as part of this specific Oracle Application Server instance, the Oracle HTTP Server.

The actual list of components shown on the Application Server page will vary depending upon the installation type you selected.

Use the **Create OC4J Instance** button to create a new OC4J instance in this application server instance. For more information, see Section 6.2.

2.3.5 Using Application Server Control to View and Manage an OC4J Instance

If you click the name of an OC4J instance on the Cluster Topology page or on the Application Server page, Enterprise Manager displays the OC4J Home page, as shown in Figure 2–3.

Figure 2–3   Managing an OC4J Instance from the OC4J Home Page
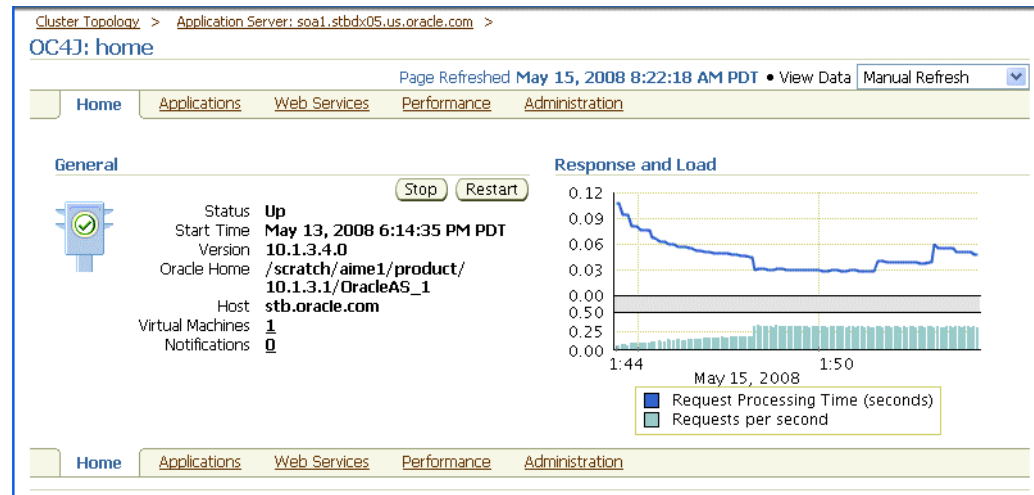
Use the OC4J Home page to obtain general information about the OC4J instance, and to view the Response and Load chart. Select a refresh interval from the **View Data** drop-down menu (for example, **30 Second Refresh**) to monitor the response and load metrics over a period of time. To start, stop, or restart the OC4J instance, navigate to the Cluster Topology page, select the instance, and click **Start**, **Stop**, or **Restart**.

The OC4J Home page and its related **Applications**, **Web Services**, **Performance**, and **Administration** pages are designed to provide you with a central, Web-based view of the OC4J instance, as well as the applications and Web services deployed to the instance.

For more information, click **Help** at any time while using the OC4J Home page. The online help provides reference information about the fields on each page, as well as links to related tasks and related documentation that can help you get started.

2.3.6 About MBeans and the Application Server Control MBean Browsers

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

MBeans are defined in the J2EE Management Specification (JSR-77), which is part of Java Management Extensions, or JMX, a set of specifications that allow standard interfaces to be created for managing applications in a J2EE environment.

You can create MBeans for deployment with an application into OC4J, enabling the application or its components to be managed and monitored through Application Server Control.

> **See Also:**   "Using MBeans in OC4J" in the *Oracle Containers for J2EE Configuration and Administration Guide*

Application Server Control provides a set of MBean browsers that allow to you browse the MBeans for an OC4J instance, a cluster, or for a selected application. You can also perform specific monitoring and configuration tasks from the MBean browser.

For more information, see the following sections:

- Viewing the System MBean Browser

- [Viewing the MBeans for a Selected Application](#)
- [Viewing the Cluster MBean Browser](#)

2.3.6.1 Viewing the System MBean Browser

To view the System MBean Browser, which contains the MBeans specific to the selected OC4J instance:

1. Navigate to the OC4J Home page for the OC4J instance.

2. Click **Administration** to display the OC4J Administration page.

3. Click the task icon in the **System Bean Browser** row of the table.

   Enterprise Manager displays the System MBean Browser. Click **Help** if you need more information about using the System Bean Browser.

   Note that the online help also provides a set of topics that provide a tour of the MBean Browser, including online demonstrations of using the MBean browser.

   > **See Also:**   "Step 1: Explore the Structure of the MBean Browser" in the Application Server Control online help

2.3.6.2 Viewing the MBeans for a Selected Application

To view the MBeans for a specific application:

1. Click **Applications** on the OC4J Home page to display the list of applications deployed to the OC4J instance.

2. Click the name of a deployed application.

3. Click **Administration** to display the Application Administration page, which contains a table listing the various administration tasks you can perform for the selected application.

4. Click the appropriate task icon in the table to browse the System MBeans associated with the selected application or to view the MBeans defined for the application.

   Enterprise Manager displays the selected MBean Browser page.

2.3.6.3 Viewing the Cluster MBean Browser

To view the MBeans associated with the cluster topology, click **Cluster MBean Browser** on the Cluster Topology page.

Enterprise Manager displays the Cluster MBean Browser, which highlights the MBeans used to manage the cluster and any groups defined within the cluster.

3

Starting and Stopping

This chapter describes procedures for starting and stopping Oracle Application Server.

It contains the following topics:

- Overview of Starting and Stopping Procedures
- Starting and Stopping Application Server Instances
- Starting and Stopping Components
- Starting and Stopping an Oracle Application Server Environment
- Starting and Stopping: Special Topics

3.1 Overview of Starting and Stopping Procedures

Oracle Application Server is a flexible product that you can start and stop in different ways, depending on your requirements. See the following sections:

- Section 3.2, "Starting and Stopping Application Server Instances"

  Use the procedures in this section when starting an instance from scratch, for example, after restarting a host, or when you want to stop your entire instance, for example, in preparation for shutting down your system.

- Section 3.3, "Starting and Stopping Components"

  Use the procedures in this section after you have started your instance and want to start or stop individual components.

- Section 3.4, "Starting and Stopping an Oracle Application Server Environment"

  This section describes how to perform an orderly shutdown of your entire environment.

3.2 Starting and Stopping Application Server Instances

This section describes how to start and stop application server instances. It contains the following topics:

- Starting a Middle-Tier Instance
- Stopping a Middle-Tier Instance

For information about stopping the entire environment, including a 10.1.4 or 10.1.2 OracleAS Infrastructure connected to the middle-tier instance, see Section 3.4.

3.2.1 Starting a Middle-Tier Instance

This section describes how to start all processes in a middle-tier instance. You can follow this procedure after you have restarted your host, or any other time you want to start up the entire instance.

To start a middle-tier instance:

1. If the middle-tier instance is associated with OracleAS Infrastructure services, such as Oracle Identity Management, make sure the services are started. See Section 3.4.3 for more information.

2. Start the middle-tier components:

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

   This command starts OPMN and all OPMN-managed processes such as Oracle HTTP Server, OC4J instances, and Application Server Control.

Alternatively, on Windows, you can start the middle tier from the Windows Start menu. For example, to start an Oracle Application Server on Windows 2000, select **Start** > **Programs** > **Oracle - *Oracle_home_name*** > **Oracle Process Manager** > **Start Oracle Process Manager**. This starts OPMN and all processes managed by OPMN.

3.2.2 Stopping a Middle-Tier Instance

This section describes how to stop all processes in a middle-tier instance. Follow this procedure when you are preparing to shut down your host, or any other time you want to stop the entire instance.

To stop a middle-tier instance, use the following command:

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

This command stops OPMN and all OPMN-managed processes such as Oracle HTTP Server, OC4J instances, and Application Server Control.

Alternatively, on Windows, you can stop the middle tier from the Windows Start menu. For example, to stop an Oracle Application Server instance on Windows 2000, select **Start** > **Programs** > **Oracle - *Oracle_home_name*** > **Oracle Process Manager** > **Stop Oracle Process Manager**. This stops OPMN and all processes managed by OPMN.

3.3 Starting and Stopping Components

You can use the following tools to start, stop, restart, and view the status of components:

- The `opmnctl` command. See Section 3.3.1.

- Application Server Control. See Section 3.3.2.

These tools are completely compatible—they both use OPMN as their underlying technology for managing processes—and can often be used interchangeably. For example, you can start a component using `opmnctl` and stop it using Application Server Control.

Although the two tools can be used interchangeably, they offer different features. With the `opmnctl` command, you can start and stop sub-processes within components, as well as the entire component. With Application Server Control, you can view

components that cannot be started or stopped, but whose status depends on other components.

3.3.1 Starting and Stopping Components Using opmnctl

You can use the `opmnctl` command-line tool to start and stop components. It is located in the following directory:

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

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

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

To start, stop, or restart the sub-process of a component:

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

To start, stop, or restart an application, such as Application Server Control:

```
opmnctl startproc application=app_name
opmnctl stopproc application=app_name
opmnctl restartproc application=app_name
```

To view the status of components and processes:

```
opmnctl status -l
```

To learn more about using opmnctl, refer to *Oracle Process Manager and Notification Server Administrator's Guide*.

3.3.2 Starting and Stopping Components Using Application Server Control

You can start, stop, restart, and view the status of components on the Application Server home page:

1. Navigate to the Cluster Topologies home page on Application Server Control.

2. In the Members section, for **View By,** select **Application Servers.**

3. In the **Select** column, select the components you want to start, stop, or restart. (Click **Expand All** to show all the components.)

4. Click **Start, Stop,** or **Restart** on the top right of the Members section.

You can also start and stop individual components on each component home page.

3.4 Starting and Stopping an Oracle Application Server Environment

This section provides procedures for starting and stopping an Oracle Application Server environment. An environment can consist of multiple OracleAS Infrastructure and middle-tier instances distributed across multiple hosts. These instances are dependent on each other and it is important to start and stop them in the proper order.

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

This section contains the following topics:

- Starting an Oracle Application Server Environment
- Stopping an Oracle Application Server Environment
- Starting 10.1.4 or 10.1.2 OracleAS Infrastructure
- Stopping 10.1.4 or 10.1.2 OracleAS Infrastructure

Note: In this chapter, OracleAS Infrastructure refers to a Release 10.1.4 or Release 2 (10.1.2) OracleAS Infrastructure.

For information about associating a 10*g* Release 3 (10.1.3.4) middle-tier instance with an OracleAS Infrastructure, see Section 6.7.

3.4.1 Starting an Oracle Application Server Environment

To start an Oracle Application Server environment:

1. Start any OracleAS Infrastructure that contains only OracleAS Metadata Repository.

    If your environment has OracleAS Infrastructure installations that contain only OracleAS Metadata Repository, start those in any order. Note that for these installation types, you only need to start OracleAS Metadata Repository. You do not need to start any processes with opmnctl and you do not need to start Application Server Control. See Section 3.4.3.

2. Start the OracleAS Infrastructure that contains Oracle Identity Management.

    If your environment uses Oracle Identity Management, start the OracleAS Infrastructure that contains Oracle Internet Directory. If this OracleAS Infrastructure contains OracleAS Metadata Repository, start that before you start Oracle Internet Directory. See Section 3.4.3.

3. Start middle-tier instances.

    Start middle-tier instances in any order. See Section 3.2.1.

3.4.2 Stopping an Oracle Application Server Environment

To stop all processes in an Oracle Application Server environment:

1. Stop middle-tier instances.

    If your environment contains middle-tier instances, stop them in any order. See Section 3.2.2.

2. Stop the OracleAS Infrastructure that contains Oracle Identity Management.

    If your environment uses Oracle Identity Management, stop the OracleAS Infrastructure that contains Oracle Internet Directory. If this OracleAS Infrastructure contains OracleAS Metadata Repository, stop that as well.

3. Stop any OracleAS Infrastructure instances that contain only OracleAS Metadata Repository.

    If your environment has OracleAS Infrastructure instances that contain only OracleAS Metadata Repository, stop those in any order. See Section 3.4.4.

3.4.3 Starting 10.1.4 or 10.1.2 OracleAS Infrastructure

If your middle tier is connected to a Release 10.1.4 or Release 2 (10.1.2) OracleAS Infrastructure instance, you can stop the Infrastructure from that instance.

This procedure applies to all OracleAS Infrastructure types:

- Oracle Identity Management and OracleAS Metadata Repository

  Follow both steps to start Oracle Identity Management and OracleAS Metadata Repository.

- OracleAS Metadata Repository only

  Follow only Step 1 to start OracleAS Metadata Repository. You do not need to perform the second step of starting Oracle Identity Management because you do not need OPMN or Application Server Control in a OracleAS Metadata Repository-only installation.

- Oracle Identity Management only

  Follow only Step 2 to start Oracle Identity Management. Make sure the OracleAS Metadata Repository that supports Oracle Identity Management (residing in another Oracle home) is already started.

To start OracleAS Infrastructure:

1. If your OracleAS Infrastructure contains OracleAS Metadata Repository, start it as follows:

   a. Set the ORACLE_HOME environment variable to the OracleAS Infrastructure Oracle home.

   b. Set the ORACLE_SID environment variable to the OracleAS Metadata Repository SID (default is `orcl`).

   c. Start the Net Listener:

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

   d. Start the OracleAS Metadata Repository instance:

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

   e. Start the Oracle Enterprise Manager 10*g* Database Control:

   ```
   (UNIX) ORACLE_HOME/bin/emctl start dbconsole
   (Windows) ORACLE_HOME\bin\emctl start dbconsole
   ```

2. If your OracleAS Infrastructure contains Oracle Identity Management, start it as follows:

   a. Start components:

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

   This command starts OPMN and all OPMN-managed processes such as DCM, Oracle HTTP Server, OC4J instances, and Oracle Internet Directory.

   b. Start Application Server Control:

   ```
   (UNIX) ORACLE_HOME/bin/emctl start iasconsole
   ```

```
(Windows) ORACLE_HOME\bin\emctl start iasconsole
```

Alternatively, on Windows, you can start the Application Server Control Console using the Windows Services control panel. The name of the service for the Application Server Control uses the following format:

```
OracleOracleHomeNameASControl
```

See Section A.1 for more information on starting Application Server Control.

Alternatively, on Windows, you can start the Infrastructure from the Programs menu: **Start** > **Programs** > **Oracle Application Server Infrastructure** - *Infra_name* > **Start** *instanceName*.

3.4.4 Stopping 10.1.4 or 10.1.2 OracleAS Infrastructure

If your middle tier is connected to a Release 10.1.4 or Release 2 (10.1.2) OracleAS Infrastructure instance, you can stop the Infrastructure from that instance.

This procedure applies to all OracleAS Infrastructure types:

- Oracle Identity Management and OracleAS Metadata Repository

  Follow both steps to stop Oracle Identity Management and OracleAS Metadata Repository.

- OracleAS Metadata Repository only

  Follow Step 2 only to stop OracleAS Metadata Repository.

- Oracle Identity Management only

  Follow Step 1 only to stop Oracle Identity Management.

To stop OracleAS Infrastructure:

1. If your OracleAS Infrastructure contains Oracle Identity Management, stop it as follows:

   a. Stop Application Server Control:

   ```
   (UNIX) ORACLE_HOME/bin/emctl stop iasconsole
   (Windows) ORACLE_HOME\bin\emctl stop iasconsole
   ```

   Alternatively, on Windows, you can stop the Application Server Control Console using the Services control panel. See Section A.1 for more information.

   b. Stop components:

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

   This command stops OPMN and all OPMN-managed processes such as DCM, Oracle HTTP Server, OC4J instances, and Oracle Internet Directory.

2. If your OracleAS Infrastructure contains OracleAS Metadata Repository, stop it as follows:

   a. Set the ORACLE_HOME environment variable to the OracleAS Infrastructure Oracle home.

   b. Set the ORACLE_SID environment variable to the OracleAS Metadata Repository SID (default is `orcl`).

   c. Stop the OracleAS Metadata Repository instance:

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

d. Stop the Net Listener:

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

e. Stop the Oracle Enterprise Manager 10*g* Database Control:

```
(UNIX) ORACLE_HOME/bin/emctl stop dbconsole
(Windows) ORACLE_HOME\bin\emctl stop dbconsole
```

Alternatively, on Windows, you can stop the Infrastructure from the Programs menu: **Start** > **Programs** > **Oracle Application Server Infrastructure** - *Infra_name* > **Stop** *instanceName*.

3.5 Starting and Stopping: Special Topics

This section contains the following special topics about starting and stopping Oracle Application Server:

- Starting and Stopping in High Availability Environments
- Enabling and Disabling Components
- Resolving OC4J Errors When Starting Multiple Instances

3.5.1 Starting and Stopping in High Availability Environments

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

- Oracle Application Server Clusters
- Oracle Application Server Cold Failover Clusters
- Oracle Application Server Disaster Recovery (includes starting and stopping the ASG component)

> **See:** *Oracle Application Server High Availability Guide* for information about starting and stopping in high-availability environments

3.5.2 Enabling and Disabling Components

You can enable and disable components using opmnctl. You cannot enable and disable components using Application Server Control.

When you disable a component, you prevent it from starting when you start the application server instance. When you enable a component, you allow it to start when you start the application server instance.

To disable a component, you edit the opmn.xml file, specifying `disabled` for the <ias-component>.

To enable a component, you edit the opmn.xml file, specifying `enabled` for the <ias-component>.

When you enable or disable components, consider the following restrictions and additional information:

- If you use the backup and recovery procedures documented in this book, you must run `bkp_restore.pl -m config` after you enable or disable components so the proper components are registered with the OracleAS Backup and Recovery Tool. See Chapter 15 for more information.

- You cannot disable or enable components that are part of Oracle Application Server Clusters.

> **See Also:**
>
> - *Distributed Configuration Management Administrator's Guide* for information about using Distributed Configuration Management to create and manage an OracleAS Clusters
>
> - "About Managing OracleAS Clusterss" in the Application Server Control Console online Help

In addition, you can create or remove OC4J instances, as described in Section 6.2.

3.5.3 Resolving OC4J Errors When Starting Multiple Instances

If you have multiple Oracle Application Server installations on one host and you start them at the same time, OPMN may return an error similar to the following:

```
<process-type id="my_OC4J_instance">
  <process-set id="default_island">
    <process id="93388820" pid="24711" status="Stopped" index="1"
     log="/disk1/oracleas/opmn/logs/OC4J~my_OC4J_instance~default_island~1"
 operation="request" result="failure">
      <msg code="-21" text="failed to restart a managed process
        after the maximum retry limit">
      </msg>
```

This error indicates that an OC4J instance (`my_OC4J_instance`) failed to start. The problem could be caused by two different Oracle homes on the same host using the same port ranges for RMI, JMS, and AJP ports, and an OC4J instance in one Oracle home trying to use the same port as an OC4J instance in another Oracle home.

For example, assume you have two Oracle Application Server installations on one host that reside in `ORACLE_HOME1` and `ORACLE_HOME2`. Each installation contains one or more OC4J instances, and each OC4J instance is assigned a port range for AJP, RMI, and JMS ports.

You can check OC4J port range assignments by examining the `opmn.xml` file in both Oracle homes:

```
ORACLE_HOME1/opmn/conf/opmn.xml
ORACLE_HOME2/opmn/conf/opmn.xml
```

In each file, locate the OC4J instance entries, which start with a line like the following:

```
<process-type id="home" module-id="OC4J" ... >
```

Within each entry, locate the RMI, JMS, and AJP port ranges, which looks like this:

```
<port id="ajp" range="12501-12600"/>
<port id="rmi" range="12401-12500"/>
<port id="jms" range="12601-12700"/>
```

Table 3–1 illustrates the problem of having the same OC4J port assignments in two Oracle homes—the AJP, RMI, and JMS port ranges in `ORACLE_HOME1` are identical to

the AJP, RMI, and JMS port ranges in ORACLE_HOME2. (Note that this example only lists the relevant lines from the opmn.xml.)

Table 3–1   Example of Identical Port Ranges in Two Oracle Homes

OC4J Port Ranges in *ORACLE_HOME1*/opmn/conf/opmn.xml	OC4J Port Ranges in *ORACLE_HOME2*/opmn/conf/opmn.xml

<pre><ias-component id="OC4J">
  ...
  <process-type id="home" ... >
    ...
    <port id="default-web-site"
     range="**12501**-**12600**" protocol="ajp"/>
    <port id="rmi" range="**12401**-**12500**"/>
    <port id="rmis" range="**12701**-**12800**"/>
    <port id="jms" range="**12601**-**12700**"/>
    ...
  </process-type>
  <process-type id="OC4J_WebCenter" ... >
    ...
    <port id="default-web-site"
     range="**12501**-**12600**" protocol="ajp"/>
    <port id="rmi" range="**12401**-**12500**"/>
    <port id="rmis" range="**12701**-**12800**"/>
    <port id="jms" range="**12601**-**12700**"/>
  </process-type></pre>
	<pre><ias-component id="OC4J">
	  ...
	   <process-type id="home" ... >
	     ...
	     <port id="default-web-site
	        range="**12501**-**12600**" protocol="ajp"/>
	     <port id="rmi" range="**12401**-**12500**"/>
	     <port id="rmis" range="**12701**-**12800**"/>
	     <port id="jms" range="**12601**-**12700**"/>
	     ...
	   </process-type>
	   <process-type id="OC4J_WebCenter" ... >
	     ...
	     <port id="default-web-site"
	      range="**12501**-**12600**" protocol="ajp"/>
	     <port id="rmi" range="**12401**-**12500**"/>
	     <port id="rmis" range="**12701**-**12800**"/>
	     <port id="jms" range="**12601**-**12700**"/>
	   </process-type></pre>

Port allocation for all OC4J instances within an Oracle Application Server instance is controlled by OPMN. So, having overlapping port ranges within a single opmn.xml file is not a problem. However, when two OPMNs on a host start processes at the same time, there is no coordination between them on port usage.

The algorithm OPMN uses to assign a port is:

1. Choose a port from the port range that is not currently marked as allocated to any processes managed by the OPMN in the local instance.

2. Before assigning the port, check to see if the port is in use by binding to it.

3. If the port is not in use (that is, OPMN could bind to it), then unbind and assign the port to a process (such as an OC4J instance) so it can bind to it, updating internal data structures with this assignment information.

In between the time that OPMN unbinds from the port and the assigned process binds to the port, it is possible for another process to bind to the port. This could be another OPMN on the host, or any other process that happens to try to bind to the same port number.

If your port range assignments are the same across Oracle homes, and you received the error shown at the beginning of this section, then probably two OPMN processes tried to bind the same port for their OC4J instances. There is no way to eliminate this problem completely (because there is a rare chance that a non-OPMN process could try to bind to the port at the same time) but you can reconfigure OPMN to reduce the chance of encountering it.

There are two options for addressing this problem:

■ Option 1: Assign Unique Port Ranges to Each Oracle Home

■ Option 2: Increase the Maximum Number of Retries for Starting OC4J Instances

Option 1: Assign Unique Port Ranges to Each Oracle Home

You can assign unique OC4J port ranges to each Oracle home, as shown in Table 3–2. Then, the OPMN in `ORACLE_HOME1` and the OPMN in `ORACLE_HOME2` will not attempt to use the same port numbers when assigning OPMN ports, and will not attempt to bind to the same port.

Table 3–2   Example of Using Unique Port Ranges in Two Oracle Homes

OC4J Port Ranges in *ORACLE_HOME1*/opmn/conf/opmn.xml	OC4J Port Ranges in *ORACLE_HOME2*/opmn/conf/opmn.xml

``
<ias-component id="OC4J">
  ...
  <process-type id="home" ... >
    ...
    <port id="default-web-site"
     range="12501-12600" protocol="ajp"/>
    <port id="rmi" range="12401-12500"/>
    <port id="rmis" range="12701-12800"/>
    <port id="jms" range="12601-12700"/>
    ...
  </process-type>
  <process-type id="OC4J_WebCenter" ... >
    ...
    <port id="default-web-site"
     range="12501-12600" protocol="ajp"/>
    <port id="rmi" range="12401-12500"/>
    <port id="rmis" range="12701-12800"/>
    <port id="jms" range="12601-12700"/>
  </process-type>
``
	``
	<ias-component id="OC4J">
	  ...
	  <process-type id="home" ... >
	    ...
	    <port id="default-web-site"
	     range="4601-4700" protocol="ajp"/>
	    <port id="rmi" range="4701-4800"/>
	    <port id="rmis" range="4901-4999"/>
	    <port id="jms" range="4801-4900"/>
	    ...
	  </process-type>
	  <process-type id="OC4J_WebCenter" ... >
	    ...
	    <port id="default-web-site"
	     range="4601-4700" protocol="ajp"/>
	    <port id="rmi" range="4701-4800"/>
	    <port id="rmis" range="4901-4999"/>
	    <port id="jms" range="4801-4900"/>
	  </process-type>
	``

To do this:

1. Choose unique port ranges for AJP, RMI, and JMS.

2. Edit *ORACLE_HOME2*/opmn/conf/opmn.xml.

3. For each OC4J instance in the file, change AJP, RMI, and JMS to use the new unique port ranges. For example:

   ```
   <port id="ajp" range="4601-4700"/>
   <port id="rmi" range="4701-4800"/>
   <port id="jms" range="4801-4900"/>
   ```

4. Save and close the file.

5. Reload OPMN:

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

Option 2: Increase the Maximum Number of Retries for Starting OC4J Instances

OPMN attempts to start processes a certain number of times before declaring failure. For process types with port ranges, if the failure to start the process is due to the process not being able bind to the assigned port number, OPMN attempts to start the process with a different port number in the specified range. You can have identical port ranges in two Oracle homes, and increase the number of times OPMN attempts to restart a process, so eventually OPMN will choose a port that works. This does not completely eliminate the problem, because there is a chance that OPMN will not find a port that works in 10 tries, but it does reduce the chance of encountering the problem.

The parameter that controls the number of retries is `retry`. The default value is 2. You can increase the parameter to a higher number, for example, 10, by following these steps in each Oracle home:

1. Edit *ORACLE_HOME*/opmn/conf/opmn.xml.

2. For each OC4J instance in the file, increase the retry value for start and restart. For example:

```
<start timeout="600" retry="10"/>
<restart timeout="720" retry="10"/>
```

3. Save and close the file.

4. Reload OPMN:

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

Table 3–3 shows an example of the opmn.xml file in two Oracle homes on the same host after the retry count has been increased to 10.

Table 3–3 Example of Increasing the Retry Count in Two Oracle Homes

OC4J Port Ranges in *ORACLE_HOME1*/opmn/conf/opmn.xml	OC4J Port Ranges in *ORACLE_HOME2*/opmn/conf/opmn.xml

<pre><ias-component id="OC4J">
  ...
  <process-type id="home" ... >
    ...
    <start timeout="600" retry="**10**"/>
    ...
    <restart timeout="720" retry="**10**"/>
    <port id="default-web-site"
     range="**12501**-**12600**" protocol="ajp"/>
    <port id="rmi" range="**12401**-**12500**"/>
    <port id="rmis" range="**12701**-**12800**"/>
    <port id="jms" range="**12601**-**12700**"/>
    ...
  </process-type>
  <process-type id="OC4J_WebCenter" ... >
    ...
    <start timeout="600" retry="**10**"/>
    ...
    <restart timeout="720" retry="**10**"/>
    <port id="default-web-site"
     range="**12501**-**12600**" protocol="ajp"/>
    <port id="rmi" range="**12401**-**12500**"/>
    <port id="rmis" range="**12701**-**12800**"/>
    <port id="jms" range="**12601**-**12700**"/>
  </process-type></pre>
	<pre><ias-component id="OC4J">
	  ...
	  <process-type id="home" ... >
	    ...
	    <start timeout="600" retry="**10**"/>
	    ...
	    <restart timeout="720" retry="**10**"/>
	    <port id="default-web-site"
	     range="**12501**-**12600**" protocol="ajp"/>
	    <port id="rmi" range="**12401**-**12500**"/>
	    <port id="rmis" range="**12701**-**12800**"/>
	    <port id="jms" range="**12601**-**12700**"/>
	    ...
	  </process-type>
	  <process-type id=""OC4J_WebCenter" ... >
	    ...
	    <start timeout="600" retry="**10**"/>
	    ...
	    <restart timeout="720" retry="**10**"/>
	    <port id="default-web-site"
	     range="**12501**-**12600**" protocol="ajp"/>
	    <port id="rmi" range="**12401**-**12500**"/>
	    <port id="rmis" range="**12701**-**12800**"/>
	    <port id="jms" range="**12601**-**12700**"/>
	  </process-type></pre>

Part II

Basic Administration

This part describes basic administration tasks.

It contains the following chapters:

4

Managing Ports

This chapter describes how to view and change Oracle Application Server port numbers. It contains the following topics:

- About Managing Ports
- Viewing Port Numbers
- Changing Middle-Tier Ports
- Changing 10.1.4 or 10.1.2 Infrastructure Ports

4.1 About Managing Ports

Many Oracle Application Server components and services use ports. 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.

Most port numbers are assigned during installation. Every component and service has an allotted port range, which is the set of port numbers that Oracle Application Server attempts to use when assigning a port. Oracle Application Server starts with the lowest number in the range and performs the following checks:

- Is the port used by another Oracle Application Server installation on the host?

  The installation may be up or down at the time; Oracle Application Server can still detect if the port is used.

- Is the port used by a process that is currently running?

  This could be any process on the host, even a non-Oracle Application Server process.

If the answer to any of the preceding questions is yes, Oracle Application Server moves to the next highest port in the allotted port range and continues checking until it finds a free port.

You can override this behavior for some ports, and specify a port number assignment during installation. To do this, you edit a template file called `staticports.ini`, and launch Oracle Universal Installer with special options.

> **See Also:** Appendix D for a complete list of allotted port ranges.
> Refer to *Oracle Application Server Installation Guide* for directions on overriding port assignments during installation with `staticports.ini`.

4.2 Viewing Port Numbers

You can view the current port numbers by using the following command:

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

You can also view the port numbers using Application Server Control. From the Cluster Topology page, select **Runtime Ports.**

4.3 Changing Middle-Tier Ports

This section provides complete instructions for changing port numbers in middle-tier instances. The instructions explain how to change the port number, and update any other components that might be affected.

> **Note:** You can change a port number to any number you want, as long as it is an unused port. You do not have to use a port in the allotted port range for the component. See Appendix D for information on allowed port ranges.

This section contains the following topics:

- Changing the OC4J Listener Port
- Changing Other OC4J Ports
- Changing the Oracle HTTP Server Listen Ports
- Changing the Oracle HTTP Server Diagnostic Port
- Changing the Java Object Cache Port
- Changing OPMN Ports (ONS Local, Request, and Remote)
- Changing the Port Tunneling Port

4.3.1 Changing the OC4J Listener Port

If you did not install Oracle HTTP Server, OC4J provides an HTTP listener. You can change OC4J listener using Application Server Control or manual steps:

- Using the Application Server Control:

  1. If Java SSO is enabled for Application Server Control, disable it:

     From the Cluster Topology page, click **Java SSO Configuration.** Then, on the Java SSO Configuration page, click **Participating Applications.** On the Participating Applications page, make sure `ascontrol` is not checked. Click **Apply.**

  2. From the Cluster Topology page, click **Runtime Ports.**

  3. Click the **Configure Port** icon for the HTTP port of the OC4J instance (by default `home`.)

  4. In the Ports section of the Server Properties page, locate the **Web Sites** table. In the table, change the port for **default-web-site.**

  5. Click **Apply.**

  6. Restart the OC4J instance:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_instance
(Windows) ORACLE_HOME\opmn\bin\opmnctl startproc process-type=OC4J_instance
```

- Using manual steps:

  1. Execute the following command. For example, if the OC4J default instance is named `home`:

     ```
     (UNIX) ORACLE_HOME/opmn/bin/opmnctl config port update
         ias-component=default_group process-type=home range=8888
         portid=default-web-site
     ```

     ```
     (Windows) ORACLE_HOME\opmn\bin\opmnctl config port update
         ias-component=default_group process-type=home range=8888
         portid=default-web-site
     ```

  2. Reload OPMN:

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

  3. Start the OC4J instance that contains the port number you changed:

     ```
     (UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_instance
     (Windows) ORACLE_HOME\opmn\bin\opmnctl startproc process-type=OC4J_instance
     ```

     For example, if you changed a port number in the `home` instance on UNIX:

     ```
     ORACLE_HOME/opmn/bin/opmnctl startproc process-type=home
     ```

4.3.2 Changing Other OC4J Ports

This section describes how to change the following OC4J port numbers:

- AJP
- JMS
- RMI
- RMIS
- IIOP
- IIOPS1 (Server only)
- IIOPS2 (Server and client)

By default, Oracle Application Server does not use a single port number for each type of OC4J port. Instead, it uses a port range for each type of OC4J port and that range is the same for all OC4J instances on the host. During run time, each OC4J instance on the host is assigned a single free port from the range. For example, if the default AJP range for every OC4J instance on a host is 12501-12600, then each OC4J instance is assigned a single free port from that range for its AJP port.

When changing an OC4J port number, you typically specify a new port range. The range may be a simple port range (12501-12600), a comma separated list of ports (12501, 12504, 12507), or a combination of both (12501-12580, 12583, 12590-12600). By default, the ranges contain 100 ports. If you specify a range that is too narrow, you may encounter problems when starting OC4J instances. The AJP and RMI port ranges are required; the others are optional.

> **Note:** Note that because the IIOP, IIOPS1, and IIOPS2 ports are not configured by default, they may not be listed in the opmn.xml file. To configure them, you must manually add them to the file.
>
> See the *Oracle Containers for J2EE Services Guide* for more information.

You can change OC4J port ranges using Application Server Control or manual steps:

- Using Application Server Control:

  1. From the Cluster Topology page, click **Runtime Ports.**

  2. Click the **Configure Port** icon for the port you want to change.

  3. In the Ports section of the Server Properties page, change the port range for the port you want to change.

  4. Click **Apply.**

  5. Navigate to the Cluster Topology page, select the OC4J instance that you modified, and click **Restart.**

  6. Click **Yes** on the confirmation page.

- Using manual steps:

  1. Open the `opmn.xml` file:

  ```
  (UNIX) ORACLE_HOME/opmn/conf/opmn.xml
  (Windows) ORACLE_HOME\opmn\conf\opmn.xml
  ```

  2. Locate the element for the OC4J instance that contains the port range you want to change. For example, if you want to change a port range for the `home` instance, locate this element:

  ```
  <process-type id="home" ...>
  ```

  3. Within the OC4J instance element, there is a `port` element for each type of port. For example, if you installed an instance using the Basic Installation option:

  ```
  <port id="default-web-site" range="8888" protocol="http"/>
  <port id="rmi" range="12401-12500"/>
  <port id="rmis" range="12701-12800"/>
  <port id="jms" range="12601-12700"/>
  <port id="iiop" range="13301-13400"/>
  <port id="iiops1" range="13401-13500"/>
  <port id="iiops2" range="13501-13600"/>
  ```

  4. Modify the range parameter for the port you want to change, and then save the file.

  5. Reload OPMN:

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

  6. Start the OC4J instance that contains the port number you changed:

  ```
  (UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_instance
  (Windows) ORACLE_HOME\opmn\bin\opmnctl startproc process-type=OC4J_instance
  ```

  For example, if you changed a port number in the `home` instance on UNIX:

```
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=home
```

4.3.3 Changing the Oracle HTTP Server Listen Ports

To change the Oracle HTTP Server Listen ports (non-SSL or SSL), you change the Oracle HTTP Server Listen directive. When you do this, there are often dependencies that must also be set. For example, if you are using OracleAS Web Cache Release 2 (10.1.2) to improve the performance of your Oracle Application Server instance, you must modify the OracleAS Web Cache origin server settings whenever you modify the Oracle HTTP Server Listen ports.

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

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

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

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

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

1. Log in as root.

2. Run the following commands in the middle-tier Oracle home:

   ```
   cd ORACLE_HOME/Apache/Apache/bin
   chown root .apachectl
   chmod 6750 .apachectl
   ```

4.3.3.2 Changing the Oracle HTTP Server Non-SSL Listen Ports

To change the Oracle HTTP Server non-SSL (HTTP) Listen port, follow the procedures in the following tasks. Note that, on a UNIX system, if you are changing the Listen port to a number less than 1024, you must first perform the steps in Section 4.3.3.1.

- Task 1: Modify the Oracle HTTP Server HTTP Listen Directive
- Task 2: Update OracleAS Web Cache
- Task 3: Change the Port Number in Oracle Content DB
- Task 4: Restart the Middle-Tier Instance

Task 1: Modify the Oracle HTTP Server HTTP Listen Directive

To change the Oracle HTTP Server HTTP Listen directive, take the following steps:

1. Open the `httpd.conf` file:

   ```
   (UNIX) ORACLE_HOME/Apache/Apache/conf/httpd.conf
   (Windows) ORACLE_HOME\Apache\Apache\conf\httpd.conf
   ```

2. Update the Listen directive with the new port number.

There may be multiple Listen directives in this file. Modify the Listen directive that is not enclosed in an SSL virtual host container. The easiest way to locate the proper Listen directive is to search the file for the old port number.

3. If the middle-tier instance is *not* using OracleAS Web Cache as a reverse proxy, update the Port directive.

   The value for Listen and Port must be the same port number. The following example changes the Listen port to 7779:

   ```
   Listen 7779
   Port 7779
   ```

   If the middle-tier instance is using OracleAS Web Cache as a reverse proxy, do not update the Port directive.

Task 2: Update OracleAS Web Cache

If the middle-tier instance is using Release 2 (10.1.2) OracleAS Web Cache as a reverse proxy, you must update OracleAS Web Cache.

For example, for a Release 2 (10.1.2) OracleAS Web Cache standalone instance, take the following steps:

1. From OracleAS Web Cache Manager for the Release 2 (10.1.2) instance, select **Origin Servers, Sites, and Load Balancing** > **Origin Servers.**

2. Select the HTTP port and click **Edit Selected.**

3. In the Edit Application Web Server dialog box, change the number in the **Port** field.

4. Click **Submit.**

5. Click **Apply Changes.**

6. Click **Restart.**

Task 3: Change the Port Number in Oracle Content DB

If your environment includes Oracle Content DB, you must change the Oracle Content DB Application Port domain property and restart the OC4J_Content instance. See the section "Changing the Oracle Content DB Port Number" in the *Oracle Content Database for Oracle WebCenter Suite Administrator's Guide* for the specific steps.

Task 4: Restart the Middle-Tier Instance

Restart the application server instance:

- On UNIX systems:

  ```
  ORACLE_HOME/opmn/bin/opmnctl stopall
  ORACLE_HOME/opmn/bin/opmnctl startall
  ```

- On Windows systems:

  ```
  ORACLE_HOME\opmn\bin\opmnctl stopall
  ORACLE_HOME\opmn\bin\opmnctl startall
  ```

4.3.3.3 Changing the Oracle HTTP Server SSL Listen Port

To change the Oracle HTTP Server SSL (HTTPS) Listen port, follow the procedures in the following tasks. Note that, on a UNIX system, if you are changing the Listen port to a number less than 1024, you must perform the steps in Section 4.3.3.1.

- Task 1: Modify the Oracle HTTP Server Listen Directive

- Task 2: Update OracleAS Web Cache

- Task 3: Re-register mod_osso

- Task 4: Change the Port Number in Oracle Content DB

- Task 5: Restart the Middle-Tier Instance

Task 1: Modify the Oracle HTTP Server Listen Directive

If you are changing the HTTPS port, change both the SSL Listen and Port directives to the new port number in the Oracle HTTP Server `ssl.conf` file, using the following steps:

1. Edit the `ssl.conf` file, located at:

    ```
    (UNIX) ORACLE_HOME/Apache/Apache/conf
    (Windows) ORACLE_HOME\Apache\Apache\conf
    ```

2. Update the SSL Listen and SSL Port directives, and the VirtualHost _default directive with the new port number, and then save the file.

    The value for Listen, Port, and VirtualHost _default must be the same port number. The following example shows the directives changed to port 4445:

    ```
    Listen 4445
    Port 4445
    <VirtualHost _default_:4445>
    ```

    Save and close the file.

3. Restart the middle-tier instance:

    - On UNIX systems:

      ```
      ORACLE_HOME/opmn/bin/opmnctl stopall
      ORACLE_HOME/opmn/bin/opmnctl startall
      ```

    - On Windows systems:

      ```
      ORACLE_HOME\opmn\bin\opmnctl stopall
      ORACLE_HOME\opmn\bin\opmnctl startall
      ```

Task 2: Update OracleAS Web Cache

If the middle-tier instance is using Release 2 (10.1.2) OracleAS Web Cache as a reverse proxy, you must update OracleAS Web Cache. Take the following steps:

1. From OracleAS Web Cache Manager for the Release 2 (10.1.2) instance, select **Origin Servers, Sites, and Load Balancing** > **Origin Servers.**

2. Select the HTTPS port and click **Edit Selected.**

3. In the Edit Application Web Server dialog box, change the number in the **Port** field.

4. Click **Submit.**

5. Click **Apply Changes.**

6. Click **Restart.**

Task 3: Re-register mod_osso

If you have enabled OracleAS Single Sign-On authentication (that is, you registered mod_osso), follow these steps to re-register mod_osso:

1. On the Identity Management host, set the environment variables ORACLE_ HOME and ORACLE_SID.

2. On the Identity Management host, run the `ssoreg` script, using the `-remote_ midtier` option. The script is located at:

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

   For example, on LINUX:

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

   The resulting configuration file (`myosso.conf` in the example) is an obfuscated osso configuration file.

3. Copy the obfuscated osso configuration file to the 10*g* Release 3 (10.1.3.4) middle-tier instance.

4. On the middle-tier host, run the following script to complete the registration:

   ```
   (UNIX) ORACLE_HOME/Apache/Apache/bin/osso1013 config_file
   (Windows) perl ORACLE_HOME\Apache\Apache\bin\osso1013 config_file
   ```

Task 4: Change the Port Number in Oracle Content DB

If your environment includes Oracle Content DB, you must change the Oracle Content DB ApplicationUseHttps domain property and restart the OC4J_Content instance. See the section "Changing the Oracle Content DB Port Number" in the *Oracle Content Database for Oracle WebCenter Suite Administrator's Guide* for the specific steps.

Task 5: Restart the Middle-Tier Instance

Restart the application server instance:

- On UNIX systems:

  ```
  ORACLE_HOME/opmn/bin/opmnctl stopall
  ORACLE_HOME/opmn/bin/opmnctl startall
  ```

- On Windows systems:

  ```
  ORACLE_HOME\opmn\bin\opmnctl stopall
  ORACLE_HOME\opmn\bin\opmnctl startall
  ```

4.3.4 Changing the Oracle HTTP Server Diagnostic Port

To change the Oracle HTTP Server Diagnostics port number:

1. Open the `dms.conf` file:

   ```
   (UNIX) ORACLE_HOME/Apache/Apache/conf/dms.conf
   (Windows) ORACLE_HOME\Apache\Apache\conf\dms.conf
   ```

2. Change the old port number to the new port number everywhere it appears in the file, and then save the file. This update includes the Listen directive, OpmnHostPort directive, Redirect directive, and the VirtualHost.

3. Restart Oracle HTTP Server:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=HTTP_Server
(UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=HTTP_Server

(Windows) ORACLE_HOME\opmn\bin\opmnctl stopproc ias-component=HTTP_Server
(Windows) ORACLE_HOME\opmn\bin\opmnctl startproc ias-component=HTTP_Server
```

4.3.5 Changing the Java Object Cache Port

To change the Java Object Cache port number in any installation type:

1. Open the `javacache.xml` file:

```
(UNIX) ORACLE_HOME/javacache/admin/javacache.xml
(Windows) ORACLE_HOME\javacache\admin\javacache.xml
```

2. Under the `<communication>` element, update the `discovery-port` parameter in the `<coordinator>` element with the new port number, and then save the file.

For example:

```
<coordinator discovery-port="7010" />
```

3. Restart all OC4J instances which contain J2EE applications that use Java Object Cache:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_instance_name
(Windows) ORACLE_HOME\opmn\bin\opmnctl restartproc process-type=OC4J_instance_
name
```

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

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

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

To change these ports:

1. Stop Application Server Control, OPMN, and all OPMN-managed processes:

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

2. Open the `opmn.xml` file:

```
(UNIX) ORACLE_HOME/opmn/conf/opmn.xml
(Windows) ORACLE_HOME\opmn\conf\opmn.xml
```

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

For example:

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

4. Start Application Server Control, OPMN, and all OPMN-managed processes:

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

4.3.7 Changing the Port Tunneling Port

To change the Port Tunneling port number:

1. Stop Application Server Control, OPMN, and all OPMN-managed processes:

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

2. Open the `opmn.xml` file:

```
(UNIX) ORACLE_HOME/opmn/conf/opmn.xml
(Windows) ORACLE_HOME\opmn\conf\opmn.xml
```

3. Under the `<ias-component id="IASPT">` element, update the range parameter in the `<port>` element with the new range. For example:

```
<port id="ajp" range="7501-7553"/>
```

Note that the port number range specified in `opmn.xml` overrides any port number specified in `iaspt.conf`. As a result, you only need to update the port number in `opmn.xml`.

4. Start Application Server Control, OPMN, and all OPMN-managed processes:

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

4.4 Changing 10.1.4 or 10.1.2 Infrastructure Ports

This section contains the following topics:

- Changing the 10.1.4 or 10.1.2 OracleAS Metadata Repository Net Listener Port

- Changing 10.1.4 or 10.1.2 Oracle Internet Directory Ports

- Changing the HTTP Server Port on a 10.1.4 or 10.1.2 Identity Management Installation

- Changing 10.1.4 or 10.1.2 OracleAS Certificate Authority Ports

4.4.1 Changing the 10.1.4 or 10.1.2 OracleAS Metadata Repository Net Listener Port

If your environment includes a 10.1.4 or 10.1.2 OracleAS Metadata Repository, and you want to change the listener port number, perform the procedure in this section.

First, determine if it is necessary to change the OracleAS Metadata Repository listener port number. If you are concerned about the fact that you have another database on your host using the same port, it is possible that the OracleAS Metadata Repository and the other database can use the same port.

The following are guidelines for port usage by multiple databases on the same host:

- Multiple Oracle9*i* and Oracle Database 10*g* databases can share the same Oracle Net listener port. If you install a OracleAS Metadata Repository on a host that contains Oracle9*i* and Oracle Database 10*g* databases, they can all use port 1521. There is no need to change the OracleAS Metadata Repository port number.

- If the other databases on your system are Oracle8*i* databases running the Net8 listener, then the OracleAS Metadata Repository must use a different port. They cannot share the same port.

> **Note:** If you want to run two listeners that use the same key value on one host, refer to Section 4.4.1.1, "Changing the KEY Value for an IPC Listener"

If you determine that you want to change the OracleAS Metadata Repository listener port, follow the procedure in this section. An OracleAS Metadata Repository may be used in several different ways. Use the following table to determine the steps that are required for changing your type of OracleAS Metadata Repository:

If the Metadata Repository is used as follows:	Follow these tasks to change its Oracle Net listener port:
■ Identity Management Repository and Product Metadata Repository ■ Registered with Oracle Internet Directory	Task 1: Stop Middle-Tier Instances
	Task 2: Change the OracleAS Metadata Repository Oracle Net Listener Port
	Task 3: Update Oracle Internet Directory
	Task 4: Update OracleAS Single Sign-On
	Task 5: Update OracleAS Certificate Authority
	Task 6: Update Application Server Control
	Task 7: Update Middle-Tier Instances
■ Identity Management Repository only ■ Registered with Oracle Internet Directory	Task 1: Stop Middle-Tier Instances
	Task 2: Change the OracleAS Metadata Repository Oracle Net Listener Port
	Task 3: Update Oracle Internet Directory
	Task 4: Update OracleAS Single Sign-On
	Task 5: Update OracleAS Certificate Authority
	Task 6: Update Application Server Control
■ Product Metadata Repository ■ Registered with Oracle Internet Directory	Task 1: Stop Middle-Tier Instances
	Task 2: Change the OracleAS Metadata Repository Oracle Net Listener Port
	Task 3: Update Oracle Internet Directory
	Task 7: Update Middle-Tier Instances

Task 1: Stop Middle-Tier Instances

Stop all middle-tier instances that use the Metadata Repository by executing the following command in each middle-tier Oracle home:

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

Task 2: Change the OracleAS Metadata Repository Oracle Net Listener Port

On the OracleAS Metadata Repository host:

1. Make sure that the ORACLE_HOME and ORACLE_SID environment variables are set.

2. If OPMN is running, stop it:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl stopall
```

```
(Windows) ORACLE_HOME\opmn\bin\opmnctl stopall
```

3. Stop the OracleAS Metadata Repository listener:

```
lsnrctl stop
```

4. 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.

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

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

Make the following changes to the file:

a. Update the PORT value in each entry that applies to OracleAS Metadata Repository.

b. Add an entry like the following:

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

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

6. Start the OracleAS Metadata Repository listener:

```
lsnrctl start
```

7. Using SQL*Plus, log in to the OracleAS Metadata Repository as the SYSTEM user with SYSDBA privileges and run the following command:

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

8. Using SQL*Plus, restart OracleAS Metadata Repository:

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

9. Start Oracle Internet Directory:

   - On UNIX systems:

     ```
     ORACLE_HOME/opmn/bin/opmnctl start
     ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=OID
     ```

   - On Windows systems:

     ```
     ORACLE_HOME\opmn\bin\opmnctl start
     ORACLE_HOME\opmn\bin\opmnctl startproc ias-component=OID
     ```

Task 3: Update Oracle Internet Directory

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

1. Start Oracle Directory Manager:

   - On UNIX, use the following command:

```
ORACLE_HOME/bin/oidadmin
```

- On Windows, navigate to Oracle Directory Manager (**Start**, **Programs**, **Oracle Application Server Infrastructure -** *Oracle_Home*, **Integrated Management Tools**, **Oracle Directory Manager**).

2. Log in to Oracle Directory Manager.

3. In the System Objects frame:

   a. Expand **Entry Management.**

   b. Expand **cn=Oracle Context.**

   c. Select the DBName for the OracleAS Metadata Repository. For example, if the DBName is the default, `orcl`, select **cn=ORCL**.

   d. On the Properties tab, update the `PORT` parameter in the **orclnetdescstring** field with the new port number.

4. Click **Apply.**

5. In the System Objects frame:

   a. Under **cn=Oracle Context**, select the DBName for the OracleAS Metadata Repository. For example, if the DBName is the default, `orcl`, select **cn=ORCL.**

   b. Expand **cn=DESCRIPTION_0.**

   c. Select **cn=ADDRESS_0.**

   d. On the Properties tab, update the `PORT` parameter in the **orclnetaddressstring** field with the new port number.

6. Click **Apply.**

7. Start OPMN in the Oracle Internet Directory Oracle home:

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

Task 4: Update OracleAS Single Sign-On

From the OracleAS Single Sign-On Oracle home:

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

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

   - On UNIX systems:

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

   - On Windows systems:

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

3. Restart OC4J:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl restartproc ias-component=OC4J
(Windows) ORACLE_HOME\opmn\bin\opmnctl restartproc ias-component=OC4J
```

Task 5: Update OracleAS Certificate Authority

If OracleAS Certificate Authority is configured in the Identity Management installation:

1. Run the following command:

```
(UNIX) ORACLE_HOME/oca/bin/ocactl updateconnection
(Windows) ORACLE_HOME\oca\bin\ocactl updateconnection
```

2. Restart OracleAS Certificate Authority:

```
(UNIX) ORACLE_HOME/oca/bin/ocactl stop
(UNIX) ORACLE_HOME/oca/bin/ocactl start

(Windows) ORACLE_HOME\oca\bin\ocactl stop
(Windows) ORACLE_HOME\oca\bin\ocactl start
```

If you are not sure if OracleAS Certificate Authority is configured, examine the Application Server Control Home page to see if it is listed in the Components section.

Task 6: Update Application Server Control

Update Application Server Control with the new port number:

1. In the Identity Management Oracle home, edit the following file:

```
(UNIX) ORACLE_HOME/sysman/emd/targets.xml
(Windows) ORACLE_HOME\sysman\emd\targets.xml
```

2. Update each occurrence of the old OracleAS Metadata Repository port number with the new port number and save the file.

   For example, update the `PORT` parameter and the `ConnectDescriptor` parameter.

3. Reload Application Server Control:

```
(UNIX) ORACLE_HOME/bin/emctl reload
(Windows) ORACLE_HOME\bin\emctl reload
```

Task 7: Update Middle-Tier Instances

In each middle-tier Oracle home that uses OracleAS Metadata Repository:

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

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

2. Check the following file:

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

   Locate the line that begins with `PlsqlDatabaseConnectString`.

   - If the line ends with `ServiceNameFormat` or `SIDFormat`, update the line with the new OracleAS Metadata Repository port number, save the file, and restart Oracle HTTP Server.

   - If the line ends with `NetServiceNameFormat`, you do not need to do anything.

3. Start the middle-tier instance:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl startall
```

```
(Windows) ORACLE_HOME\opmn\bin\opmnctl startall
```

4.4.1.1 Changing the KEY Value for an IPC Listener

It is not possible to run two listeners at the same time that are configured to use the same KEY value in their IPC protocol address. By default, the OracleAS 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 OracleAS 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 (make sure 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
    ```

4.4.2 Changing 10.1.4 or 10.1.2 Oracle Internet Directory Ports

You can associate a 10*g* Release 3 (10.1.3.4) middle-tier instance with Release 10.1.4 or Release 2 (10.1.2) Identity Management Services.

When you change the Oracle Internet Directory HTTP or HTTPS port on an Identity Management installation, you must update any middle-tier instance that uses the Identity Management installation.

The following tasks describe how to update the Oracle Internet Directory port number, including updating other components in the Infrastructure and updating the middle-tier instances that use the port:

■   Task 1: Prepare the Middle-Tier Instances

■   Task 2: Prepare the Infrastructure Instances

■   Task 3: Change the Oracle Internet Directory Port

■   Task 4: Reconfigure OracleAS Certificate Authority

■   Task 5: Restart the Identity Management Instance

■   Task 6: Update the Middle-Tier Instances to Use the New Port Number

Task 1: Prepare the Middle-Tier Instances

For each 10*g* Release 3 (10.1.3.4) middle-tier instance that uses Identity Management, make sure the instance is started.

If it is not started, start all processes, using the following command:

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

Task 2: Prepare the Infrastructure Instances

Prepare the Infrastructure instances by taking these steps:

1. Make sure that Identity Management and its associated OracleAS Metadata Repository are started on the Infrastructure whose port number you are changing.

2. If any middle-tier instance uses a different OracleAS Metadata Repository for its product metadata, make sure that repository is started. In short, make sure all Metadata Repositories in your environment are started.

Task 3: Change the Oracle Internet Directory Port

Change the Oracle Internet Directory port by taking these steps:

1. On the Oracle Internet Directory host:

    a. Create a file named `mod.ldif` with the following contents. You can create the file in any directory.

        For HTTP:

        ```
        dn: cn=configset0, cn=osdldapd, cn=subconfigsubentry
        changetype: modify
        replace: orclnonsslport
        orclnonsslport: new_nonssl_port_number
        ```

        For HTTPS:

        ```
        dn: cn=configset0, cn=osdldapd, cn=subconfigsubentry
        changetype: modify
        replace: orclsslport
        orclsslport: new_ssl_port_number
        ```

    b. Run the following command:

        For the HTTP (non-SSL) port:

        ```
        ldapmodify -D "cn=orcladmin" -w password -p oid_port -f mod.ldif
        ```

        For the HTTPS (SSL) port:

        ```
        ldapmodify -D "cn=orcladmin" -w password -p oid_port -U SSLAuth -f mod.ldif
        ```

        Note that `oid_port` is the old Oracle Internet Directory port number. If you are changing the HTTPS port, provide the additional `-U` argument to specify the SSL authentication mode. Use one of the following values for `SSLAuth`: 1 for no authentication required; 2 for one-way authentication required; 3 for two-way authentication required.

2. On the Oracle Internet Directory host, stop the entire instance that contains Oracle Internet Directory, as well as Application Server Control:

    ■ On UNIX systems:

        ```
        ORACLE_HOME/bin/emctl stop iasconsole
        ORACLE_HOME/opmn/bin/opmnctl stopall
        ```

    ■ On Windows systems:

        ```
        ORACLE_HOME\bin\emctl stop iasconsole
        ORACLE_HOME\opmn\bin\opmnctl stopall
        ```

3. Perform this step in the Oracle Internet Directory Oracle home. If you have OracleAS Metadata Repository installed in other Oracle homes that are registered

with this Oracle Internet Directory, perform this step in each of those Oracle homes as well.

a. Open the `ldap.ora` file:

```
(UNIX) ORACLE_HOME/ldap/admin/ldap.ora
(Windows) ORACLE_HOME\ldap\admin\ldap.ora
```

b. Modify the following line to contain the new port number, and then save the file:

```
DIRECTORY_SERVERS=(myhost.myco.com:non_ssl_port:ssl_port)
```

c. Open the `ias.properties` file:

```
(UNIX) ORACLE_HOME/config/ias.properties
(Windows) ORACLE_HOME\config\ias.properties
```

d. Change the value of `OIDport` (for an HTTP port change) or `OIDsslport` (for an HTTPS port change) to the new port number, and then save the file.

4. On the Oracle Internet Directory host, start the instance that contains Oracle Internet Directory, and start Application Server Control:

   ■ On UNIX systems:

```
ORACLE_HOME/opmn/bin/opmnctl startall
ORACLE_HOME/bin/emctl start iasconsole
```

   ■ On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl startall
ORACLE_HOME\bin\emctl start iasconsole
```

5. Perform this step in the OracleAS Single Sign-On Oracle home:

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

b. Run the following command in the OracleAS Single Sign-On Oracle home:

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

Task 4: Reconfigure OracleAS Certificate Authority

Perform this task if you are using OracleAS Certificate Authority:

1. If OracleAS Certificate Authority is running in a different Oracle home, do the following steps in the OracleAS Certificate Authority Oracle home:

a. Open the `ias.properties` file:

```
(UNIX) ORACLE_HOME/config/ias.properties
(Windows) ORACLE_HOME\config\ias.properties
```

b. Change the value of `OIDport` (for an HTTP port change) or `OIDsslport` (for an HTTPS port change) to the new port number, and then save the file.

2. Update OracleAS Certificate Authority with the new Oracle Internet Directory port number by running the following command in the OracleAS Certificate Authority Oracle home:

```
(UNIX) ORACLE_HOME/oca/bin/ocactl changesecurity -server_auth_port portnum
```

```
(Windows) ORACLE_HOME\oca\bin\ocactl changesecurity -server_auth_port portnum
```

In the example, `portnum` is the OracleAS Certificate Authority Server Authentication Virtual Host (SSL) port; the default is 6600.

> **See Also:** *Oracle Application Server Certificate Authority Administrator's Guide*, Release 2 (10.1.2), for more information

Task 5: Restart the Identity Management Instance

Restart the Identity Management instance:

- On UNIX systems:

```
ORACLE_HOME/bin/emctl stop iasconsole
ORACLE_HOME/opmn/bin/opmnctl stopall
ORACLE_HOME/opmn/bin/opmnctl startall
ORACLE_HOME/bin/emctl start iasconsole
```

- On Windows systems:

```
ORACLE_HOME\bin\emctl stop iasconsole
ORACLE_HOME\opmn\bin\opmnctl stopall
ORACLE_HOME\opmn\bin\opmnctl startall
ORACLE_HOME\bin\emctl start iasconsole
```

Task 6: Update the Middle-Tier Instances to Use the New Port Number

On each middle-tier instance that uses the Identity Management installation, run the Change Identity Management Services wizard and start the instance:

1. Using the Application Server Control Console, navigate to the OC4J Home page for the middle-tier instance.

2. Click **Administration.**

3. In the Task Name column of the table, expand **Security** if it is not already expanded. Then, in the Identity Management row, click the **Go to Task** icon.

4. On the Identity Management page, click **Change.**

5. Follow the steps in the wizard for supplying the new Identity Management information. See Section 6.7 for more information.

6. When the operation is finished, click **Restart** to restart the OC4J instance, and click **YES** in the confirmation page.

You must also update any Release 2 (10.1.2) middle-tier instance that uses Identity Management. For information about updating Release 2 (10.1.2) middle-tier instances, see the "Managing Ports" chapter in the Release 2 (10.1.2) *Oracle Application Server Administrator's Guide*.

4.4.3 Changing the HTTP Server Port on a 10.1.4 or 10.1.2 Identity Management Installation

This section describes how to change the Oracle HTTP Server HTTP or HTTPS Listen port on a 10.1.4 or 10.1.2 Identity Management installation. When you change this port number, you also effectively change the OracleAS Single Sign-On port number. This means you must update any middle-tier instance that uses the OracleAS Single Sign-On port.

The following tasks describe how to update the Oracle HTTP Server port number on Identity Management, including updating other components in the Infrastructure and updating the middle-tier instances that use the port:

- Task 1: Prepare the Middle-Tier Instances
- Task 2: Prepare the Infrastructure Instances
- Task 3: Modify the Oracle HTTP Server Listen and Port Directives
- Task 4: Enable Oracle HTTP Server to Run as Root for Ports Less Than 1024 (UNIX Only)
- Task 5: Update Application Server Control
- Task 6: Update OracleAS Single Sign-On
- Task 7: Re-register mod_osso
- Task 8: Update Oracle Delegated Administration Services
- Task 9: Update OracleAS Certificate Authority
- Task 10: Restart the Identity Management Instance
- Task 11: Restart OracleAS Certificate Authority
- Task 12: Update the Middle-Tier Instances to Use the New Port Number

Task 1: Prepare the Middle-Tier Instances

For each 10*g* Release 3 (10.1.3.4) middle-tier instance that uses Identity Management, make sure the instance is started.

If it is not started, start all processes, using the following command:

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

Task 2: Prepare the Infrastructure Instances

Prepare the Infrastructure by taking the following steps:

1. Make sure that Identity Management and its associated OracleAS Metadata Repository are started on the Infrastructure whose port number you are changing.

2. If any middle-tier instance uses a different Metadata Repository for its product metadata, make sure the repository is started. In short, make sure all Metadata Repositories in your environment are started.

Task 3: Modify the Oracle HTTP Server Listen and Port Directives

If you are changing the HTTP port, change both the Listen and Port directives to the new port number in the Oracle HTTP Server `httpd.conf` file. You can perform this task using Application Server Control or manual steps on the Release 2 (10.1.2) Identity Management instance.

- Using Application Server Control on the Identity Management instance:

  1. Navigate to the Application Server Home page and click **Ports.**

  2. On the Ports page, locate the Oracle HTTP Server Listen port and click the icon in the **Configure** column.

  3. On the Server Properties page:

    – Enter the new port number in the **Default Port** field. This is for the Port directive.

–   Enter the new port number in the **Listening Port** column. This is for the Listen directive. There may be more than one listening port listed. The only way to tell which is the non-SSL Listen port is to choose the one with the old non-SSL Listen port value.

4. At the bottom of the page, click **Apply.**

5. On the Confirmation page, click **No,** you would not like to restart now.

■   Using manual steps:

1. Open the `httpd.conf` file:

    ```
    (UNIX) ORACLE_HOME/Apache/Apache/conf/httpd.conf
    (Windows) ORACLE_HOME\Apache\Apache\conf\httpd.conf
    ```

2. Update the non-SSL Listen and Port directives with the new port number, and then save the file.

    The value for Listen and Port must be the same port number. The following example changes the directives to port 7779:

    ```
    Listen 7779
    Port 7779
    ```

    There may be multiple Listen and Port directives in this file. Modify the Listen and Port directives that are not enclosed in an SSL virtual host container. The easiest way to locate the proper Listen and Port directives is to search the file for the old port number.

3. Run the following command:

    ```
    (UNIX) ORACLE_HOME/dcm/bin/dcmctl updateConfig -ct ohs
    (Windows) ORACLE_HOME\dcm\bin\dcmctl updateConfig -ct ohs
    ```

If you are changing the HTTPS port, change both the SSL Listen and Port directives to the new port number in the Oracle HTTP Server `ssl.conf` file. You must do this using the following manual steps:

1. Edit the `ssl.conf` file, located at:

    ```
    (UNIX) ORACLE_HOME/Apache/Apache/conf/ssl.conf
    (Windows) ORACLE_HOME\Apache\Apache\conf\ssl.conf
    ```

2. Update the SSL Listen and SSL Port directives with the new port number, and then save the file.

    The value for Listen and Port must be the same port number. The following example changes the directives to port 4445:

    ```
    Listen 4445
    Port 4445
    ```

    Save and close the file.

3. Run the following command:

    ```
    (UNIX) ORACLE_HOME/dcm/bin/dcmctl updateConfig -ct ohs
    (Windows) ORACLE_HOME\dcm\bin\dcmctl updateConfig -ct ohs
    ```

Task 4: Enable Oracle HTTP Server to Run as Root for Ports Less Than 1024 (UNIX Only)

Perform this task if you are changing the port to a value less than 1024 on UNIX.

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

1. Log in as root.

2. Run the following commands in the Infrastructure Oracle home:

```
cd ORACLE_HOME/Apache/Apache/bin
chown root .apachectl
chmod 6750 .apachectl
```

Task 5: Update Application Server Control

Update Application Server Control with the new port number:

1. Open the `targets.xml` file:

```
(UNIX) ORACLE_HOME/sysman/emd/targets.xml
(Windows) ORACLE_HOME\sysman\emd\targets.xml
```

2. Update each occurrence of the old Oracle HTTP Server Listen port number with the new port number, and then save the file.

   Depending on your configuration, this file may not contain any occurrences of the Oracle HTTP Server Listen port, or it may contain many occurrences. The Listen port may occur as a parameter on its own, or it may be part of a URL. The easiest way to edit this file is to search for all occurrences of the old Oracle HTTP Server Listen port number, and replace them with the new port number.

3. Reload Application Server Control:

```
(UNIX) ORACLE_HOME/bin/emctl reload
(Windows) ORACLE_HOME\bin\emctl reload
```

Task 6: Update OracleAS Single Sign-On

Perform this task if OracleAS Single Sign-On is configured to use the Oracle HTTP Server HTTP Listen port in the installation where you are changing the port.

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

2. Run one or both of the following commands in the OracleAS Single Sign-On Oracle home:

   To change the non-SSL port:

```
(UNIX) ORACLE_HOME/sso/bin/ssocfg.sh http hostname new_non_ssl_port_number
(Windows) ORACLE_HOME\sso\bin\ssocfg.bat http hostname new_non_ssl_port_number
```

   To change the SSL port:

```
(UNIX) ORACLE_HOME/sso/bin/ssocfg.sh https hostname new_ssl_port_number
(Windows) ORACLE_HOME\sso\bin\ssocfg.bat https hostname new_ssl_port_number
```

   In the examples:

   - *hostname* is the host on which OracleAS Single Sign-On is running.

   - *new_non_ssl_port_number* is the new non-SSL Oracle HTTP Server Listen port number.

- *new_ssl_port_number* is the new SSL Oracle HTTP Server Listen port number.

Task 7: Re-register mod_osso

Re-register `mod_osso` as follows:

1. Set environment variables:

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

   - On Windows systems, set the path, for example: `PATH=%PATH%;%ORACLE_HOME%\bin;%ORACLE_HOME%\lib`.

2. If you are changing the Oracle HTTP Server Listen port, re-register `mod_osso` to take care of the default partner applications by using the following command in the Identity Management Oracle home:

   On UNIX systems:

   ```
   ORACLE_HOME/sso/bin/ssoreg.sh
   -oracle_home_path identity_management_oracle_home
   -site_name identity_management_hostname:new_port_number
   -config_mod_osso TRUE
   -mod_osso_url mod_osso_url
   ```

   On Windows systems:

   ```
   ORACLE_HOME\sso\bin\ssoreg.bat
   -oracle_home_path identity_management_oracle_home
   -site_name identity_management_hostname:new_port_number
   -config_mod_osso TRUE
   -mod_osso_url mod_osso_url
   ```

   For example, if you want to change the Oracle HTTP Server Listen port to 7779 on host `myhost` on UNIX:

   ```
   $ORACLE_HOME/sso/bin/ssoreg.sh
   -oracle_home_path /disk1/oracleas
   -site_name myhost:7779
   -config_mod_osso TRUE
   -mod_osso_url http://myhost.mydomain:7779
   ```

3. If you are changing the Oracle HTTP Server SSL Listen port, perform the following steps:

   a. Re-register `mod_osso` with the new port number by running the following command in the middle-tier Oracle home:

   On UNIX systems:

   ```
   ORACLE_HOME/sso/bin/ssoreg.sh
   -oracle_home_path identity_management_oracle_home
   -site_name identity_management_hostname:new_port_number
   -config_mod_osso TRUE
   -update_mode MODIFY
   -remote_midtier
   -config_file path/osso-https.conf
   -mod_osso_url mod_osso_url
   ```

   On Windows systems:

```
ORACLE_HOME\sso\bin\ssoreg.bat
-oracle_home_path identity_management_oracle_home
-site_name identity_management_hostname:new_port_number
-config_mod_osso TRUE
-update_mode MODIFY
-remote_midtier
-config_file path\osso-https.conf
-mod_osso_url mod_osso_url
```

For example, if you want to change the Oracle HTTP Server SSL Listen port to 4445 on `myhost` on UNIX:

```
$ORACLE_HOME/sso/bin/ssoreg.sh
-oracle_home_path /disk1/oracleas
-site_name myhost:4445
-config_mod_osso TRUE
-update_mode MODIFY
-remote_midtier
-config_file $ORACLE_HOME/Apache/Apache/conf/osso/osso-https.conf
-mod_osso_url http://myhost.mydomain:7778
```

> **See Also:** *Oracle Application Server Single Sign-On Administrator's Guide*, Release 2 (10.1.2) for more information on registering `mod_osso`

b. Copy the obfuscated osso configuration file to the 10*g* Release 3 (10.1.3.4) middle-tier instance.

c. On the middle-tier host, run the following script to complete the registration:

```
(UNIX) ORACLE_HOME/Apache/Apache/bin/osso1013 config_file
(Windows) perl ORACLE_HOME\Apache\Apache\bin\osso1013 config_file
```

d. Edit the `mod_osso.conf` file, which is located at:

```
(UNIX) ORACLE_HOME/Apache/Apache/conf/mod_osso.conf
(Windows) ORACLE_HOME\Apache\Apache\conf\mod_osso.conf
```

In the `mod_osso.conf` file, comment the following directive, if you have not previously done so:

On UNIX systems:

```
LoadModule osso_module libexec/mod_osso.so
```

On Windows systems:

```
LoadModule osso_module modules\ApacheModuleOsso.dll
```

e. In the `httpd.conf` file, which is found in the same (`conf`) directory, add the directive that you just commented in the preceding step (if you have not previously done so). In a default setup, place the directive right after:

```
LoadModule wchandshake_module libexec/mod_wchandshake.so
```

4. Restart the Oracle HTTP Server:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
(Windows) ORACLE_HOME\opmn\bin\opmnctl restartproc process-type=HTTP_Server
```

5. If you have configured or modified any additional partner applications, you must also re-register those.

> **See Also:** *Oracle Application Server Single Sign-On Administrator's Guide*, Release 2 (10.1.2) for more information on registering mod_osso

Task 8: Update Oracle Delegated Administration Services

If you have Oracle Delegated Administration Services configured, and Oracle Delegated Administration Services uses the new port number, follow these steps to update the Oracle Delegated Administration Services URL entry in Oracle Internet Directory.

You can find out what port Oracle Delegated Administration Services uses with the following command:

```
ldapsearch -h oid_host -p oid_port -D "cn=orcladmin"
-w "password" -b "cn=OperationURLs,cn=DAS,cn=Products,cn=OracleContext"
-s base "objectclass=*" orcldasurlbase
```

To update Oracle Delegated Administration Services:

1. Create a file named mod.ldif with the following contents (you can create the file in any directory):

   ```
   dn:cn=OperationURLs,cn=DAS,cn=Products,cn=OracleContext
   changetype:modify
   replace:orcldasurlbase
   orcldasurlbase:http://hostname:new_http_port_number/
   ```

   Note the slash at the end of the orcldasurlbase URL.

2. Run the following command:

   ```
   ldapmodify -D cn=orcladmin -w password -p oid_port -f mod.ldif
   ```

Task 9: Update OracleAS Certificate Authority

If you are using OracleAS Certificate Authority:

1. Re-register OracleAS Certificate Authority with the OracleAS Single Sign-On server by running the following command in the OracleAS Certificate Authority Oracle home:

   ```
   (UNIX) ORACLE_HOME/oca/bin/ocactl changesecurity -server_auth_port portnum
   (Windows) ORACLE_HOME\oca\bin\ocactl changesecurity -server_auth_port portnum
   ```

   In the example, portnum is the OracleAS Certificate Authority Server Authentication Virtual Host (SSL) port; the default is 6600.

   > **See Also:** *Oracle Application Server Certificate Authority Administrator's Guide*, Release 2 (10.1.2)

2. If OracleAS Certificate Authority is located in a different Oracle home than the OracleAS Single Sign-On server, restart Oracle HTTP Server and the oca instance in the OracleAS Certificate Authority Oracle home:

   - On UNIX systems:

     ```
     ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=HTTP_Server
     ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=oca
     ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=HTTP_Server
     ORACLE_HOME/opmn/bin/opmnctl startproc process-type=oca
     ```

   - On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl stopproc ias-component=HTTP_Server
ORACLE_HOME\opmn\bin\opmnctl stopproc process-type=oca
ORACLE_HOME\opmn\bin\opmnctl startproc ias-component=HTTP_Server
ORACLE_HOME\opmn\bin\opmnctl startproc process-type=oca
```

Task 10: Restart the Identity Management Instance

Restart the Identity Management instance:

- On UNIX systems:

  ```
  ORACLE_HOME/bin/emctl stop iasconsole
  ORACLE_HOME/opmn/bin/opmnctl stopall
  ORACLE_HOME/opmn/bin/opmnctl startall
  ORACLE_HOME/bin/emctl start iasconsole
  ```

- On Windows systems:

  ```
  ORACLE_HOME\bin\emctl stop iasconsole
  ORACLE_HOME\opmn\bin\opmnctl stopall
  ORACLE_HOME\opmn\bin\opmnctl startall
  ORACLE_HOME\bin\emctl start iasconsole
  ```

Task 11: Restart OracleAS Certificate Authority

If OracleAS Certificate Authority is configured in this instance, restart it:

```
(UNIX) ORACLE_HOME/oca/bin/ocactl start
(Windows) ORACLE_HOME\oca\bin\ocactl start
```

Task 12: Update the Middle-Tier Instances to Use the New Port Number

Now that you have changed the Oracle HTTP Server port on the Identity Management installation, you must update all middle-tier instances to use the new port number. Note that in these steps, you do not need to explicitly specify the HTTP or HTTPS port number; the Change Identity Management wizard retrieves the port number internally.

For each 10*g* Release 3 (10.1.3.4) middle-tier instance that uses Identity Management:

1. Using Application Server Control, navigate to the OC4J Home page for the middle-tier instance.

2. Click **Administration.**

3. In the Task Name column of the table, expand **Security** if it is not already expanded. Then, in the Identity Management row, click the **Go to Task** icon.

4. On the Identity Management page, click **Change.**

5. Enter the current information for Oracle Internet Directory. See Section 6.7 for more information.

6. Click **OK.**

7. When the operation is finished, click **Restart** to restart the OC4J instance, and click **YES** in the confirmation page.

You must also update any Release 2 (10.1.2) middle-tier instance that uses Identity Management. For information about updating Release 2 (10.1.2) middle-tier instances, see the "Managing Ports" chapter in the Release 2 (10.1.2) *Oracle Application Server Administrator's Guide.*

4.4.4 Changing 10.1.4 or 10.1.2 OracleAS Certificate Authority Ports

This section describes how to change the following port numbers:

- OracleAS Certificate Authority Server Authentication Virtual Host (SSL)
- OracleAS Certificate Authority Mutual Authentication Virtual Host (SSL)

To change either of these port numbers:

1. Open the `ocm_apache.conf` file in the Oracle home of the Infrastructure that contains OracleAS Certificate Authority:

   ```
   (UNIX) ORACLE_HOME/Apache/Apache/conf/ocm_apache.conf
   (Windows) ORACLE_HOME\Apache\Apache\conf\ocm_apache.conf
   ```

   a. Modify the `Server` or `Mutual` port, or both, and then save the file.

      Note that each port number is listed in the file in two places:

      - As a Listen directive
      - As a default virtual host

      The easiest way to find these is to search for the old port number.

   b. Run the following command:

      ```
      (UNIX) ORACLE_HOME/dcm/bin/dcmctl updateConfig -ct ohs
      (Windows) ORACLE_HOME\dcm\bin\dcmctl updateConfig -ct ohs
      ```

2. Run the following command (make sure your ORACLE_HOME environment variable is set first):

   ```
   sqlplus oca/oca_admin_password @$ORACLE_HOME/oca/sql/ocaportchg
   ```

   a. Enter the Server Authentication Only port when prompted. If you do not want to change this port number, enter the old port number.

   b. Enter the Mutual Authentication port when prompted. If you do not want to change this port number, enter the old port number.

3. Re-register OracleAS Certificate Authority with the OracleAS Single Sign-On server by executing the following command in the OracleAS Certificate Authority Oracle home:

   ```
   (UNIX) ORACLE_HOME/oca/bin/ocactl changesecurity -server_auth_port portnum
   (Windows) ORACLE_HOME\oca\bin\ocactl changesecurity -server_auth_port portnum
   ```

   In the example, `portnum` is the OracleAS Certificate Authority Server Authentication Virtual Host (SSL) port; the default is 6600.

   > **See Also:** *Oracle Application Server Certificate Authority Administrator's Guide*

4. Restart Oracle HTTP Server:

   ```
   (UNIX) ORACLE_HOME/opmn/bin/opmnctl restartproc type=ohs
   (Windows) ORACLE_HOME\opmn\bin\opmnctl restartproc type=ohs
   ```

5. Restart the OracleAS Certificate Authority OC4J instance:

   ```
   (UNIX) ORACLE_HOME/opmn/bin/opmnctl restartproc type=oc4j instancename=oca
   (Windows) ORACLE_HOME\opmn\bin\opmnctl restartproc type=oc4j instancename=oca
   ```

6. Start Oracle Application Server Certificate Authority:

```
(UNIX) ORACLE_HOME/oca/bin/ocactl start
(Windows) ORACLE_HOME\oca\bin\ocactl start
```

5

Managing Log Files

Oracle Application Server 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. This chapter describes how to view and manage log files to assist in monitoring system activity and in diagnosing system problems.

It contains the following topics:

- Listing and Viewing Log Files with Application Server Control

- Understanding Oracle Application Server Logging

- Diagnosing Problems and Correlating Messages

- Advanced Logging Topics

5.1 Listing and Viewing Log Files with Application Server Control

With Application Server Control, you can list and search log files across Oracle Application Server components. You can view log files from Application Server Control pages or download a log file to your local client and view the log files using another tool.

This section covers the following:

- Viewing Log Files

- Listing Log Files for Components

- Searching Log Files and Viewing Messages

- Using Regular Expressions with Search

5.1.1 Viewing Log Files

You can view the log files for a specific component with Application Server Control:

1. Navigate to the Application Server Home page, then select the **Logs** link. The Log Files page is displayed.

2. You can choose to view application logs, diagnostic logs, management system logs, or Web services log, or all types of logs by selecting a option from the **View** menu.

3. In the table, expand items until you see a list of log files for the component. For example, expand **Components,** expand **HTTP Server** and **Diagnostic Logs.** The log files are listed in the table, as shown in Figure 5–1.

Figure 5–1 Enterprise Manager Log Files Page

4. For any particular log file, click **View.** The log file text is displayed.

5.1.2 Listing Log Files for Components

You can list the log files for individual components, some components, or all components by using Application Server Control. To list the log files, perform the following steps:

1. Navigate to the Application Server Home page, then select the **Logs** link. The Log Files page is displayed.

2. You can choose to view application logs, diagnostic logs, management system logs, or Web services log, or all types of logs by selecting them from the **View** menu.

3. To view all components, select **Components** in the table. To view some components, expand the Components item, then expand items under Components until you reach the level you want. Then, select those components.

4. Click **Search** to list the log files for the selected components.

5. On the Search Logs page, which is displayed when the search returns, click **Show Selected Log Files.** The Search Logs page shows the names of the log files.

5.1.3 Searching Log Files and Viewing Messages

You can filter the search for log files by certain log file attributes by using the Search Logs page of Application Server Control.

Take the following steps:

1. Navigate to the Application Server Home page, then select the **Logs** link. The Log Files page is displayed.

2. Select the type of log file by selecting it in the **View** menu.

3. To view the log files of specific components, select them in the table.

4. Click **Search.** The Search Logs page is displayed.

5. Specify the Search criteria, such as **Date Range** or **Message Type**.

6. Click **Advanced Search Options** to display the Filter by Log Message Fields table.

7. Select a field from the **Log Message Field** list.

8. Click **Add Row** to add a row for the selected log message field.

9. Enter the desired search value in the **Value** field.

10. To indicate this is a regular expression, click **Regular Expression.** (See Section 5.1.4 for information about regular expressions.)

11. If you want to select additional fields with values, click **Add Another Row** and enter additional values.

12. Click **Search** to perform the search. When the search returns, the **Results** section shows log files with matching fields, as shown in Figure 5–2.

Figure 5–2   Log Search Results Section

13. To view a log entry, click **Show** in the **Details** column of the Results area on the Search Logs page.

    The details of the error message are displayed. This information includes the Component Name and ID, Message Level, Message Type, Message Text, Module ID, and optionally the Execution Context ID (ECID).

5.1.4 Using Regular Expressions with Search

Regular expression matching is applied when you select the check box in the Regular Expression field on the Search Logs page. You can specify a regular expression for the **Message Text** field and for the **Filter by Log Message Fields** table. Using a regular expression in a search enables you to enter a pattern description to match strings for a search.

The search uses the Apache Jakarta regular expression engine, which uses "*" for a string of characters, "?" for a single character, and supports boundary matches,

including "^" for a match only at the beginning of an entry, and "$" for a match only at the end of an entry, and special characters, including "\t" for Tab, "\n" for newline, "\r" for return, and "\f" for form feed.

> **See Also:** http://jakarta.apache.org/regexp for more information on supported regular expressions

5.2 Understanding Oracle Application Server Logging

This section provides information about message formats and log file naming and explains how to configure logging options for components. This section covers the following topics:

- Understanding Log File Formats and Naming
- Configuring Component Logging Options

5.2.1 Understanding Log File Formats and Naming

Log files of Oracle Application Server components use either text-based formatting or Oracle Diagnostic Logging (ODL).

Using ODL, log file naming and the format of the contents of log files conforms to an Oracle standard and the diagnostic messages are written in XML. Some Oracle Application Server components do not use ODL, and write their diagnostic messages using a component-specific text format. Other components support ODL, but do not enable ODL by default.

Regardless of the format, ODL or text-based, of the messages that are stored in log files, you can view log files using Application Server Control, 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).

This section covers the following topics:

- ODL Message Formatting and ODL Log File Naming
- Log File Message Formats by Component

> **Note:** Some Oracle Application Server components do not support ODL. Other components support ODL, but do not enable ODL by default.

5.2.1.1 ODL Message Formatting and ODL Log File Naming

When Oracle Application Server components run and produce ODL messages, the messages are written to diagnostic log files using XML format. Each ODL message includes a HEADER element containing fields with information about the message, optionally a CORRELATION_DATA element containing information to assist in correlating messages across components, and a PAYLOAD element containing the message text, including optional arguments and associated values.

Using ODL, Oracle Application Server components write diagnostic log files to a logging directory and determine the names for logging directories using a component-specific naming convention.

See Also:

- Section 5.4.1, "Understanding ODL Messages and ODL Log Files"
- Section 5.3.1, "Correlating Messages Across Log Files and Components"

5.2.1.2 Log File Message Formats by Component

Table 5–1 lists the supported message formats for each Oracle Application Server component. Several components optionally support ODL format, where ODL is not the default format.

Table 5–1   Diagnostic Message Format by Component

Component	Default Format	ODL Support	Location[1]
Application Server Control	Text	No	*ORACLE_HOME*/j2ee/home/log/ascontrol.log
			ORACLE_HOME/j2ee/home/log/home_default_group-1/
Content DB: node log	Text	No	*ORACLE_HOME*/content/log/*domain_namenode_name.log*
Content DB: application logs	Text	Yes	*ORACLE_HOME*/j2ee/OC4J_ Content/application-deployments/Content/OC4J_ Content_default_group_1/application.log
HTTP Server	Text	Yes	*ORACLE_HOME*/Apache/Apache/logs/error_log.*time*
BPEL Process Manager	Text	No	*ORACLE_HOME*/j2ee/*instance_ name*/application-deployments/orabpel/*instance_ group_process*/application.log
Enterprise Service Bus	Text	No	*ORACLE_HOME*/j2ee/*instance_ name*/application-deployments/esb-dt/*instance_ group_process*/application.log
			ORACLE_HOME/j2ee/*instance_ name*/application-deployments/esb-rt/*instance_ group_process*/application.log
OC4J *instance_name*	Text	Yes	*ORACLE_HOME*/j2ee/*instance_name*/log/*instance_group_ process*/
			ORACLE_HOME/j2ee/*instance_ name*/application-deployments/*application_ name*/application.log
OC4J *instance_name*	ODL	Yes	*ORACLE_HOME*/j2ee/*instance_name*/log/*instance_group_ process*/oc4j/log.xml
OPMN	Text	No	*ORACLE_HOME*/opmn/logs
			ORACLE_HOME/opmn/logs/*component_type~*...
Port Tunneling	Text	No	*ORACLE_HOME*/iaspt/logs
TopLink	Text	No	Specified with the log path configuration option in the TopLink installation directory, for example: config/toplink.xml
Universal Installer	Text	No	*ORACLE_HOME*/cfgtoollogs
WebCenter: application logs	Text	Yes	*ORACLE_HOME*/j2ee/OC4J_ WebCenter/application-deployments/*type*/OC4J_ WebCenter_default_group_1/application.log
Web Services Manager	Text	No	*ORACLE_HOME*/j2ee/*instance_name/application.*log

[1] Locations are shown in UNIX format. Invert the slashes for Windows format.

5.2.2 Configuring Component Logging Options

Administrators can configure logging options to manage and limit the logging information that Oracle Application Server components generate and save.

For example, to configure logging options for OC4J components using Java logging, you modify the `j2ee-logging.xml` file, which is installed in the *ORACLE_HOME*/j2ee/home/config directory. See the chapter, "Logging in OC4J" in the *Oracle Containers for J2EE Configuration and Administration Guide* for more information.

The logging configuration options for components include:

- Specifying log file names and pathnames: Most Oracle Application Server components let you specify the directory for storing diagnostic log files. Specifying the diagnostic logging directory allows administrators to manage system and network resources.

- Limiting log file size: As Oracle Application Server components run and generate diagnostic messages, the size of the log files increases. Oracle Application Server components use one of several strategies to deal with log file size. Some components allow log files to keep increasing in size; in this case, it is the administrator's responsibility to monitor and clean up the log files. Other components, including OC4J, let you specify configuration options that limit how much log file data is collected and saved.

- Using log file archiving: Certain Oracle Application Server components let you specify configuration options to control the size of diagnostic logging directories. This lets you determine a maximum size for the directories containing a component's log files. When the maximum size is reached, older logging information is deleted before newer logging information is saved.

- Setting component logging levels: Certain Oracle Application Server components, including Oracle HTTP Server, allow administrators to configure logging levels. By configuring logging levels, the number of messages saved to diagnostic log files can be reduced. For example, you can set the logging level so that the system only reports and saves critical messages.

> **See Also:**
>
> - Section A.4, "Configuring Logging for Application Server Control"
> - Oracle Application Server component documentation for information on setting logging configuration options

5.3 Diagnosing Problems and Correlating Messages

Generally, administrators and others view log file data to diagnose, monitor, and search for component errors or problems that may cause component errors. Application Server Control supports a unified architecture and provides cross-component tools that can assist you in these tasks.

This section covers the following topics:

- Correlating Messages Across Log Files and Components
- Diagnosing Component Problems

5.3.1 Correlating Messages Across Log Files and Components

Certain Oracle Application Server components provide **message correlation** information for diagnostic messages. Message correlation information helps those

viewing diagnostic messages determine relationships between messages across components. The Execution Context ID (ECID) is a globally unique identifier associated with a thread of execution. The ECID helps you to use log file entries to correlate messages from one application or across application server components. By searching 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.

When you view an entry on the Log Message Details page in Application Server Control, if the ECID field is available, it displays the Execution Context ID as a link. Selecting the **Execution Context ID** link shows you all the diagnostic messages in the Log Repository with the same ECID.

You can use the ECID to track requests as they move through Oracle Application Server.

The ECID takes the following format:

`request_id, sequence_number`

- The `request_id` is a unique string that is associated with each request.

- The `sequence_number` represents the hop number of the request, as it passes through Oracle Application Server (or through the component).

  For example, Oracle HTTP Server assigns an initial sequence number of 0 to a request. After that, the sequence number is incremented as the request moves through Oracle Application Server components.

Table 5–2 lists the Oracle Application Server components that provide message correlation information (using an ECID), and specifies if a component supports message correlation, but it is not enabled by default.

Table 5–2   Oracle Application Server Components Supporting Message Correlation

Component	Support for Message Correlation
OC4J	Supports message correlation.
HTTP Server	Supports message correlation.

5.3.2 Diagnosing Component Problems

When an Oracle Application Server component has a problem, you can isolate and determine the cause of the problem by viewing the diagnostic messages. The following general techniques can assist you in accomplishing this task:

- Search for errors or warnings related to the problem

- Correlate the errors across components

- Correlate the errors across a time interval

- Perform component-based analysis

5.4 Advanced Logging Topics

This section covers the following topics:

- Understanding ODL Messages and ODL Log Files

- Component Diagnostic Log File Registration
- Configuring Components to Produce ODL Messages
- Managing OC4J Redirected stderr and stdout Files
- Configuration Issue for Log Files

5.4.1 Understanding ODL Messages and ODL Log Files

This section covers the following topics:

- ODL Message Contents
- ODL Log File Rotation and Naming

5.4.1.1 ODL Message Contents

Using ODL, diagnostic messages are written to log files using XML format and each message includes a HEADER element containing information about the message, optionally a CORRELATION_DATA element containing information to assist in correlating messages across components, and a PAYLOAD element containing the message text including optional arguments and associated values.

Example 5–1 shows a sample ODL format message that includes the optional CORRELATION_DATA element.

Example 5–1   Sample ODL Message Content

```
<MESSAGE>
  <HEADER>
    <TSTZ_ORIGINATING>2006-10-19T12:52:16.821-07:00</TSTZ_ORIGINATING>
    <COMPONENT_ID>j2ee</COMPONENT_ID>
    <MSG_TYPE TYPE="ERROR"></MSG_TYPE>
    <MSG_LEVEL>1</MSG_LEVEL>
    <HOST_ID>sta.oracle.com</HOST_ID>
    <HOST_NWADDR>146.87.8.203</HOST_NWADDR>
    <MODULE_ID>security</MODULE_ID>
    <THREAD_ID>10</THREAD_ID>
    <USER_ID>oracle</USER_ID>
  </HEADER>
  <CORRELATION_DATA>
    <EXEC_CONTEXT_ID><UNIQUE_ID>146.87.8.203:41990:1161287536821:0</UNIQUE_
ID><SEQ>0</SEQ></EXEC_CONTEXT_ID>
  </CORRELATION_DATA>
  <PAYLOAD>
    <MSG_TEXT>           [RealmLoginModule] authentication failed</MSG_TEXT>
  </PAYLOAD>
</MESSAGE>
```

Table 5–3 describes the contents of an ODL message header. For any given component that produces ODL format messages, the optional header fields may not be present in the generated diagnostic messages.

Table 5–3   ODL Format Message Header Fields

Header Field Name	Description	Required
COMPONENT_ID	The product or component ID for the component that originated the message.	Required
HOST_ID	The DNS host network ID.	Optional
HOST_NWADDR	The IP or other network address for the originating host.	Optional
HOSTING_CLIENT_ID	The ID of the client or security group to which the message relates.	Optional
MODULE_ID	The ID for the module that originated the message.	Optional
MSG_GROUP	The name of the group to which the message belongs, for purposes of selecting similar messages.	Optional
MSG_ID	The message ID. The message ID uniquely identifies the message.	Optional
MSG_LEVEL	An integer value that qualifies the message type (MSG_TYPE). Lower level values are for higher severity errors. Possible values are 1 through 32.	Optional
MSG_TYPE	The type of the message. Possible values are: INTERNAL_ERROR, ERROR, WARNING, NOTIFICATION, TRACE, UNKNOWN. If MSG_TYPE is included, the TYPE attribute is required when MSG_TYPE is included in the message header.	Required
ORG_ID	The organization ID for the originating component. This is usually the domain name for the organization.	Optional
PROCESS_ID	The process ID for the process, or execution unit associated with the message. Java components may use this field to specify the process ID and the thread ID, or only the thread ID.	Optional
TSTZ_NORMALIZED	The timestamp normalized for clock drift across hosts. This field is used when the diagnostic message is copied to a repository in a different hosts.	Optional
TSTZ_ORIGINATING	The timestamp with local time zone. This specifies the date and time when the message was generated.	Required
USER_ID	The User ID associated with the message.	Optional

5.4.1.2 ODL Log File Rotation and Naming

Using ODL provides the following benefits:

- The capability to limit the total amount of diagnostic information saved.

- Older segment files are removed and newer segment files are saved in chronological fashion.

- Components can remain active, and do not need to be shutdown, when diagnostic logging files are cleaned.

Using ODL, Oracle Application Server components write diagnostic log files to a logging directory. Components determine the names for logging directories using a component-specific naming convention.

An **ODL log** is a set of log files that includes the current ODL log file, typically named log.xml, 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, log.xml. When the log file reaches the rotation point, it is renamed and a new log file, log.xml is created. (You specify the rotation point, by specifying the maximum ODL segment size, and, for some OC4J logs, the rotation time and rotation frequency using component-specific configuration options.)

> **Note:** Some Oracle Application Server components, in particular
> Oracle HTTP Server, do not support the ODL log file naming
> mechanism that this section describes. In Oracle HTTP Server, ODL
> diagnostic messages are written to a file, `log.xml`, that does not have
> a configurable size limit.

Segment files are created when the ODL log file `log.xml` reaches the rotation point.
That is, the `log.xml` is renamed to `logn.xml`, where *n* is an integer, and a new
`log.xml` file is created when the component generates new diagnostic messages.

Size-Based Log Rotation

To limit the size of the ODL log, components use a configuration option specifying 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.

> **Note:** The most recent segment file is never deleted.

For example, when the maximum directory size is reached, with the starting segment
file named `log9872`, the following files could be present in the log file directory:

```
File                   Size

log.xml                10002
log9872.xml            15000
log9873.xml            15000
log9874.xml            15000
log9875.xml            15000
log9876.xml            15000
```

In this case, when `log.xml` fills up, `log9872.xml` is removed and `log.xml` is
moved to the new file `log9877.xml`. New diagnostic messages then are written to a
new `log.xml`.

For example, to specify the maximum ODL segment size and maximum directory size
for an OC4J application named petstore, you would add the following entry to the file
ORACLE_HOME/j2ee/*instance_
name*/application-deployments/petstore/orion-application.xml:

```
<log>
<odl path="../log/petstore/" max-file-size="1000" max-directory-size="10000" />
</log>
```

For OC4J components that are configured in the j2ee-logging.xml file, you can specify
a rotation time and rotation frequency, in addition to a maximum segment size and
directory size.

Time-Based Log Rotation

You specify the following properties in the <log_handler> element:

- baseRotationTime: (Optional.) The base time for the rotation. The format for the
  base time can be any of the following:

  - `hh:mm`, for example, 04:20. This format uses the local timezone.

  - `yyyy-MM-dd`, for example, 2006-08-01. This format uses the local timezone.

- yyyy-MM-ddThh:mm, for example 2006-08-01T04:20. This format uses the local timezone.

- yyyy-MM-ddThh:mm:ss.sTZD, where TZD is the timezone indicator. TZD can be Z, indicating UTC, or {+|-}hh:mm. For example, 2006-03-01T04:20:00-08:00 represents March 1, 2006 4:20:00 in US Pacific Standard Time timezone.

If you do not specify baseRotationTime, the default value is Jan. 1, 1970, 00:00 UTC.

■ rotationFrequency: The frequency of the rotation, in minutes. In addition, you can specify one of the following values: hourly, daily, weekly.

You specify these properties in the following file:

```
ORACLE_HOME/j2ee/instance_name/config/j2ee-logging.xml
```

For example, to specify that the log files are rotated every day at 4:00AM local time, or when they reach 2000000 bytes in size, use the following:

```
<log_handler name="h1" class="oracle.core.ojdl.logging.ODLHandlerFactory">
    <property name="path" value="log"/>
    <property name="baseRotationTime" value="04:00"/>
    <property name="rotationFrequency" value="daily"/>
    <property name="maxFileSize" value=" 2000000"/>
</log_handler>
```

5.4.2 Component Diagnostic Log File Registration

Application Server Control reads Oracle Application Server component diagnostic registration files to determine names, locations, and additional configuration information about diagnostic log files. The following directory contains the diagnostic log file registration files:

```
ORACLE_HOME/j2ee/instance/applications/ascontrol/ascontrol/WEB-INF/config/registration
```

Oracle Application Server components may have multiple registration files in the configuration registration directory.

The format for the registration files includes an Oracle Application Server component ID, and the extension .xml. Table 5–4 lists the Oracle Application Server components and their associated IDs.

> **Note:** Components are responsible for creating the component diagnostic registration files. Normally, Oracle Application Server administrators should not modify these files.

Table 5–4   Component IDs for Diagnostic Log File Configuration

Component Name	Component ID
Enterprise Manager	EM
Oracle HTTP Server	OHS
OC4J	OC4J
OPMN	OPMN
Port Tunneling	IASPT

Table 5–4   (Cont.) Component IDs for Diagnostic Log File Configuration

Component Name	Component ID
Oracle Content DB	CONTENT
Oracle TopLink	TOPLINK
Oracle Universal Installer	OUI
Web Services	WEBSERVICES

5.4.3 Configuring Components to Produce ODL Messages

This section covers the following topics:

- Configuring Oracle HTTP Server to Produce ODL Messages
- Configuring OC4J to Produce ODL Messages

Table 5–5 lists the Oracle Application Server components that support ODL messages but that generate text messages by default. By making configuration changes, you can configure these components to produce ODL messages. (Table 5–1 lists the Oracle Application Server components that produce ODL messages.)

Table 5–5   Components with Configuration Options for Supporting ODL

Component	Default Format	ODL Support	Location[1]
HTTP Server	Text	Yes	*ORACLE_HOME*/Apache/Apache/logs
OC4J Instance	Text	Yes	Applications deployed into OC4J:
			ORACLE_HOME/j2ee/application-deployments/*application_name*/application.log
			Note: the following OC4J log files use ODL by default:
			OC4J components using Java logging, OPMN-managed:
			ORACLE_HOME/j2ee/*instance_name*/log/*instance_group_process*/oc4j/log.xml
			OC4J components using Java logging, standalone OC4J:
			ORACLE_HOME/j2ee/*instance_name*/log/oc4j/log.xml
			See "Logging in OC4J" in *Oracle Containers for J2EE Configuration and Administration Guide* for a complete list of log files.

[1]   Locations are shown in UNIX format. Invert the slashes for Windows format.

5.4.3.1 Configuring Oracle HTTP Server to Produce ODL Messages

To configure Oracle HTTP Server to produce ODL messages, perform the following steps:

1. Add a directory named `oracle` where the Oracle HTTP Server ODL messages will be stored. Create the directory as a subdirectory of the following:

```
(UNIX) ORACLE_HOME/Apache/Apache/logs
(Windows) ORACLE_HOME\Apache\Apache\logs
```

2. Modify the `httpd.conf` file to set the value of the OraLogMode and OraLogSeverity directives. The file is located in the following directory:

```
(UNIX) ORACLE_HOME/Apache/Apache/conf/httpd.conf
(Windows) ORACLE_HOME\Apache\Apache\conf\httpd.conf
```

For example:

```
OraLogMode oracle
OraLogSeverity NOTIFICATION
```

3. Restart the middle-tier instance.

> **See Also:** *Oracle HTTP Server Administrator's Guide* for details on using the OraLogMode and OraLogSeverity directives

5.4.3.2 Configuring OC4J to Produce ODL Messages

To enable ODL logging, you add a new element, the <odl> element, within the <log> element in any of the OC4J log file configuration files. For example, to enable ODL logging for an application named petstore, you would add the following entry to the file *ORACLE_HOME*/j2ee/*instance_name*/application-deployments/petstore/orion-application.xml:

```
<log>
<odl path="../log/petstore/" max-file-size="1000" max-directory-size="10000" />
</log>
```

> **See Also:** The chapter "Logging in OC4J" in *Oracle Containers for J2EE Configuration and Administration Guide* for more information, including a list of configuration files

5.4.3.3 Directing Log4j Logs into the OC4J Logging System

The log4j framework is an open source project, sponsored by the Apache Software Foundation. The framework provides an efficient and flexible API to support run-time logging operations for Java applications. It enables developers to insert log statements into their code, incorporating messages at different levels of alarm as desired. Log4j also enables system administrators to separately define the level of logging they want to see from the application at run time, without requiring changes to the supplied application code.

Features of log4j allow you to enable logging at run time without having to modify the application binary file. Statements can remain in shipped code without incurring significant performance cost. Logging is controlled through a configuration file without requiring changes to the application binary.

For information about how to install the log4j library and configure it for use with OC4J, see "Redirecting log4j Messages for an Application to the OC4J Log" in the *Oracle Containers for J2EE Configuration and Administration Guide*.

5.4.4 Managing OC4J Redirected stderr and stdout Files

You can set run-time options for OC4J log files that specify that redirected `stderr` and `stdout` log files be rotated when the files reach a specific size or at the particular time of day. You can also specify the maximum number of log files to keep as archives. The following example specifies that the `stdout` log file will be rotated at 1:30 AM each day and that the maximum number of log files is 10:

```
java -Dstdstream.rotatetime=1:30 -Dstdstream.filenumber=10 -jar oc4j.jar -out
d:\logs\oc4j.out
```

> **See Also:** *Oracle Containers for J2EE Configuration and Administration Guide* for more information

5.4.5 Configuration Issue for Log Files

The Logs link in Application Server Control gives you an integrated view of many Oracle Application Server component log files. However, certain log files are only available at the component level. Oracle Application Server components use the following directory to make their log files visible to Application Server Control:

```
ORACLE_HOME/j2ee/home/applications/ascontrol/ascontrol/WEB-INF/config/registration
```

Some Oracle Application Server component log files are not exposed through Application Server Control pages.

Part III

Advanced Administration

This part describes advanced administration tasks that involve reconfiguring Oracle Application Server.

It contains the following chapters:

6

Reconfiguring Application Server Instances

When you installed Oracle Application Server, you chose an installation type and specified if the instance should be part of a cluster. After installation, you may want to add additional Oracle Application Server instances, add or delete OC4J instances, cluster instances, use OracleAS Web Cache as a reverse proxy, or use Identity Management with your middle-tier installation. This chapter describes how to make these types of changes.

It contains the following topics:

- Creating and Using Additional Application Server Instances
- Adding and Deleting OC4J Instances
- Configuring Cluster Topologies
- Configuring 10.1.2 OracleAS Web Cache as a Reverse Proxy
- Configuring Oracle Application Server 10.1.2 with Oracle Application Server 10.1.3
- Configuring Instances to Use OC4J Java Single Sign-On
- Configuring Instances to Use 10.1.4 or 10.1.2 Oracle Identity Management
- Disabling and Enabling Anonymous Binds

If you have disabled anonymous binds in Oracle Internet Directory, you must enable them before you make configuration changes. See Section 6.8, "Disabling and Enabling Anonymous Binds" for more information.

6.1 Creating and Using Additional Application Server Instances

This section provides instructions for creating using additional application server instances and includes the following topics:

- Overview of Multiple Application Server Instances
- Installing the Script to Create Application Server Instances
- Creating Application Server Instances
- Starting and Stopping Processes On Application Server Instances
- Removing Application Server Instances
- Changing the Default OC4J Setup
- Discovering Application Server Instances in Enterprise Manager Grid Control

6.1.1 Overview of Multiple Application Server Instances

By default, when you install Oracle Application Server, an ORACLE_HOME is installed and configured and an application server instance is created within the ORACLE_HOME.

Using a single Oracle Application Server installation (ORACLE_HOME), you can create multiple application server instances. The instances can be local to ORACLE_HOME, in a different directory on the same host as ORACLE_HOME, or distributed across hosts on the network.

An application server instance uses the binary files of ORACLE_HOME, but contains configuration files that are specific to the instance. For this reason, an application server instance is also referred to as a configuration home (CONFIG_HOME). Each CONFIG_HOME can contain an HTTP server instance, and any number of OC4J instances including a default OC4J instance. In addition, Oracle Process Manager and Notification (OPMN) server support is also included in each CONFIG_HOME to start and stop an instance's processes.

Figure 6–1 shows a conceptual view of application server instances and shows the relationship between ORACLE_HOME and CONFIG_HOME.

Figure 6–1   Conceptual View of Multiple Application Server Instances
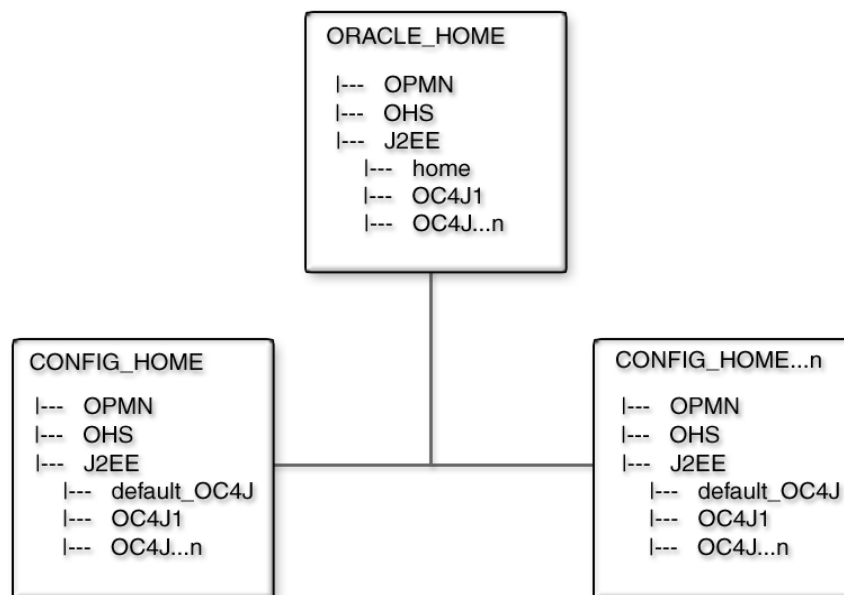
Application server instances offer a flexible deployment model and can support many possible deployment scenarios. Application server instances are also easier to administer and install, because each instance is not dependent on its own ORACLE_HOME.

You can create additional application server instances using an automation script. The script copies all the configuration files required for an application server instance, including OPMN, Oracle HTTP Server, and OC4J configuration files. Then, the script updates configuration files as required (for example, assigning port values and environment variables). Lastly, the script provides support for managing application server instances using Application Server Control and Enterprise Manager Grid Control.

6.1.2 Installing the Script to Create Application Server Instances

Before you can create application server instances, you must run the installPatch script.

Take the following steps:

1. Change to the following directory:

   ```
   $ORACLE_HOME/confighomes/bin
   ```

2. Run the installPatch script. For example:

   ```
   ./installPatch ORACLE_HOME
   ```

The script copies needed files to your Oracle home.

6.1.3 Creating Application Server Instances

To create additional application server instances, you use the `createASinstance` script. The script is located in the `ORACLE_HOME/confighomes/bin` directory and automates the tasks required to create application server instances. Those tasks include:

- Copying OPMN, Oracle HTTP Server, and OC4J configuration files, as required
- Configuring preselected fixed ports
- Updating environment variables and configuration files
- Creating the default OC4J instance, as required
- Instantiating the Oracle HTTP Server instance, as required
- Adding support for Enterprise Manager Grid Control-based management

After you run the script, you can add additional OC4J instances to an application server instance, using the `createinstance` script as described in Section 6.2.

> **Note:** The `createASinstance` script uses a password to create the default OC4J instance. The password is defined in the `ORACLE_HOME/confighomes/conf/defpwd` file and must be the same as the password that was used to create `ORACLE_HOME` during installation. If the password for `ORACLE_HOME` is different than the password defined in the `defpwd` file, update the `defpwd` file before running the `createASinstance` script.

The `createASinstance` script's usage is as follows:

```
createASinstance ORACLE_HOME ORACLE_CONFIG_HOME PORT_POOL_INDEX command
```

The following `command` options are available:

```
-full [-isNotAdmin] [-oc4jName name] [-apacheRoot path] [-repository path]
      [-startall] [-opmnSnippet file]
-oc4j [-isNotAdmin] [-oc4jName name] [-repository path] [-startall]
      [-opmnSnippet file]
-ohs  [-apacheRoot path] [-repository path] [-startall]
-opmn [-repository path] [-startall]
```

Table 6–1 describes the arguments. Examples are included after the table.

Table 6–1   Arguments for createASinstance Script

Argument	Description
ORACLE_HOME	The location where the Oracle Application Server binary files are installed.
ORACLE_CONFIG_HOME	The location where the application server instance will be installed. The application server instance can be installed at any location and does not have to be located within the ORACLE_HOME directory.
	If the location is on a remote host, you must have permission to read and write to the directory.
PORT_POOL_INDEX	A port pool index number, which determines what port numbers are used for the instance's OPMN local, remote, and request ports as well as HTTP ports (if applicable). Valid pool index values can be any number from 1-100. Each application server instance should be associated with a different port pool index.
	The index value specified is added to the base port values listed in the ORACLE_HOME/confighomes/conf/portlist.ini file to arrive at new port values for the instance. For example, the base port value for the OC4J HTTP port is 6310. An index value of 1 increases the base port value to 6311, which is then assigned to the instance's OC4J HTTP port value.
-full	Creates an application server instance that contains an OPMN instance, an Oracle HTTP Server instance, and a default OC4J instance. The OC4J default setup uses the AJP protocol, has the Application Server Control started, and is Oracle HTTP Server routable unless these items are overridden by their respective command arguments.
-oc4j	Creates an application server instance that contains an OPMN instance and a default OC4J instance. The OC4J default setup uses the HTTP protocol, has the Application Server Control started, but is not Oracle HTTP Server routable.
-ohs	Creates an application server instance that contains an OPMN instance and an Oracle HTTP Server instance.
-opmn	Creates an application server instance that contains only an OPMN instance.
-isNotAdmin	Indicates that the default OC4J instance should not start Application Server Control. See Section A.6 for information about managing the active Application Server Control.
-oc4jName *name*	The name for the default OC4J instance. The default name, if no name is provided, is default_oc4j.
-apacheRoot *path*	Creates an Oracle HTTP Server 2.0 instance. Oracle HTTP Server 2.0 is available from the Companion CD and must already be installed. The value of this argument is the path to the top-level directory where Oracle HTTP Server 2.0 is installed. The Oracle HTTP Server bundled with the application server is used by default if this argument is not specified.

Table 6–1   (Cont.) Arguments for createASinstance Script

Argument	Description
-repository *path*	Optional. An alternate directory path where the repository file (`confighomes.lst`) will be written. This file is used by Enterprise Manager Grid Control to discover the new application server instance. The default path is `ORACLE_HOME/confighomes` if no path is specified. An alternate directory is required if you do not have write permissions on `ORACLE_HOME` or when `ORACLE_HOME` is on an NFS-mounted drive and the application server instance is created on a different host than the NFS-mounted `ORACLE_HOME` host. See Section 6.1.7, "Discovering Application Server Instances in Enterprise Manager Grid Control" for more information on using the repository file.
-startall	Optional. Starts OPMN and its managed processes after creating the application server instance. OPMN must be manually started if this argument is not specified. See Section 6.1.4, "Starting and Stopping Processes On Application Server Instances" for more information about manually starting OPMN.
-opmnSnippet *file*	Optional. Specifies a different OPMN snippet file that will be used to create the default OC4J instance. The `ORACLE_HOME/confighomes/templates/config/oc4j_home.xml` file is used by default if this argument is not specified. The snippet file can be edited to change many default OC4J preferences before the instance is created. See Section 6.1.6, "Changing the Default OC4J Setup" for more information.

Examples for Creating Application Server Instances

The examples in this section demonstrate some common uses of the `createASinstance` script.

The following example uses the Application Server installation at `/shiphomes/10131_PROD/` to create a new AS instance under the `/as_instance/cfghomes/home1` directory. A default OC4J instance is created with the default name `default_oc4j` and contains an Application Server Control instance. The application server instance's ports are assigned based on the port pool index number `1`.

```
cd $ORACLE_HOME/confighomes/bin
./createASinstance /shiphomes/10131_PROD/ /as_instance/cfghomes/home1 1 -oc4j
```

The next example uses the Application Server installation at `/shiphomes/10131_PROD/` to create a new AS instance under the `/as_instance/cfghomes/home2` directory. A default OC4J instance is created with the default name `default_oc4j` and contains an Application Server Control instance. An Oracle HTTP Server instance is created and is routable to the OC4J instance using the AJP protocol. Ports are assigned based on the port pool index number `2`. Lastly, all processes on the application server instance are automatically started.

```
cd $ORACLE_HOME/confighomes/bin
./createASinstance /shiphomes/10131_PROD/ /as_instance/cfghomes/home2 2 -full
-startall
```

6.1.4 Starting and Stopping Processes On Application Server Instances

OPMN must be used to manually start and stop processes on each application server instance using the instance's `opmn/bin/opmnctl` script. The script acts as a wrapper for the `opmnctl` script located in the `ORACLE_HOME` instance and uses the same

arguments. Refer to the "Using OPMN" chapter of the *Oracle Process Manager and Notification Server Administrator's Guide* for more information on starting and stopping processes using the opmnctl script.

6.1.4.1 Starting Processes on Application Server Instances

To use OPMN to start managed processes on application server instances, issue the following command on each instance:

```
(UNIX) ORACLE_CONFIG_HOME/opmn/bin/opmnctl startall
(Windows) ORACLE_CONFIG_HOME\opmn\bin\opmnctl.bat startall
```

On Windows, the following items must be completed before starting processes on each application server instance. In addition, the ORACLE_HOME OPMN instance must be manually started.

- The Oracle Process Manager Service must be stopped. If the service is currently started, use the Windows Services Administrative tool to stop the service (**Start** -> **Control Panel** -> **Administrator Tools** -> **Services**).

> **Note:** Multiple application server instances cannot be used if the Oracle Process Manager service is used. Make sure the service is stopped and configured not to start automatically in the future.

- The ORACLE_HOME\opmn\bin\oracle.key file must be renamed.

- An ORACLE_HOME environment variable must be set to the location of the product installation directory. The environment variable is used when manually starting OPMN in ORACLE_HOME. For example:

```
set ORACLE_HOME=c:\product\shiphomes\oc4j\10131_PROD\
ORACLE_HOME\opmn\bin\opmnctl.bat startall
```

6.1.4.2 Stopping Processes On Application Server Instances

To use OPMN to stop managed processes on application server instances, issue the following command on each instance:

```
(UNIX) ORACLE_CONFIG_HOME/opmn/bin/opmnctl stopall
(Windows) ORACLE_CONFIG_HOME\opmn\bin\opmnctl.bat stopall
```

6.1.5 Removing Application Server Instances

To remove application server instances, use the removeASinstance script. The script is located in the ORACLE_HOME/confighomes/bin directory. The script removes all files associated with CONFIG_HOME and removes the instance from the Enterprise Manager Grid Control repository file.

The scripts usage is as follows:

```
removeASinstance ORACLE_HOME ORACLE_CONFIG_HOME [-repository path][-force]
```

Table 6–2 describes the arguments. Examples are included after the table.

Table 6–2   Arguments for the removeASinstance Script

Argument	Description
ORACLE_HOME	The location where the Oracle Application Server binary files are installed.

Table 6–2   (Cont.) Arguments for the removeASinstance Script

Argument	Description
ORACLE_CONFIG_HOME	The application server instance to be removed.
-repository *path*	Optional. The directory path where the repository file (confighomes.lst) is located. This file is used by Enterprise Manager Grid Control to discover application server instances. The file is updated to remove the application server instance's reference. The default path is ORACLE_HOME/confighomes if no path is specified.
	See Section 6.1.7, "Discovering Application Server Instances in Enterprise Manager Grid Control" for more information on using the repository file.
-force	Optional. Shuts down OPMN and its managed processes. If this argument is not specified, OPMN must be manually stopped before using the removeASinstance script.

Example for Removing Application Server Instances

The following example demonstrates how to remove an application server instance located at /as_instance/cfghomes/home1. All processes on the instance are fist manually stopped:

```
cd/as_instance/cfghomes/home1/opmn/bin
./opmnctl stopall
cd $ORACLE_HOME/confighomes/bin
./removeASinstance /shiphomes/10131_PROD/ /as_instance/cfghomes/home1
```

6.1.6 Changing the Default OC4J Setup

The createASinstance script uses an OPMN snippet file to create the default OC4J instance's <process-type> node in the opmn.xml file of the application server instance. The default OPMN snippet files is named oc4j_home.xml and is located in the ORACLE_HOME/confighomes/templates/config/ directory.

A new snippet file can be created and specified as the value of the -opmnSnippet argument when using the script. This allows the default OC4J instance to be preconfigured for a specific environment and minimizes the amount of configuration required after the default OC4J instance is created. Refer to the *Oracle Process Manager and Notification Server Administrator's Guide* for more information on the elements allowed in the opmn.xml file. The default OC4J setup is shown in the following example.

```
<opmn xmlns="http://www.oracle.com/ias-instance">
   <process-type id="home" module-id="OC4J" status="enabled">
      <module-data>
         <category id="start-parameters">
            <data id="java-options" value="-server -XX:MaxPermSize=128M -ms512M
               -mx1024M -XX:AppendRatio=3 -Djava.security.policy=
               $ORACLE_HOME/j2ee/home/config/java2.policy -Djava.awt.headless=true
               -Dhttp.webdir.enable=false"/>
         </category>
         <category id="stop-parameters">
            <data id="java-options" value="-Djava.security.policy=
               $ORACLE_HOME/j2ee/home/config/java2.policy -Djava.awt.headless=true
               -Dhttp.webdir.enable=false"/>
         </category>
      </module-data>
      <start timeout="600" retry="2"/>
```

```
                        <stop timeout="120"/>
                        <restart timeout="720" retry="2"/>
                        <port id="default-web-site" range="12501-12600" protocol="ajp"/>
                        <port id="rmi" range="12401-12500"/>
                        <port id="rmis" range="12701-12800"/>
                        <port id="jms" range="12601-12700"/>
                        <process-set id="default_group" numprocs="1"/>
            </process-type>
</opmn>
```

> **Note:** The snippet file is only used if the `-full` or `-oc4j` arguments are used to create a default OC4J instance.

6.1.7 Discovering Application Server Instances in Enterprise Manager Grid Control

By default, you manage Oracle Application Server instances with Application Server Control. With additional Application Server instances, you manage them with Enterprise Manager Grid Control. The instances must first be discovered using Enterprise Manager Grid Control.

> **Note:** An Enterprise Manager Grid Control Deployment Procedure can be implemented to create and discover application server instances at the same time. For detailed information, refer to the Enterprise Manager Grid Control documentation:
>
> http://www.oracle.com/technology/documentation/oem.html

To discover application server instances:

1. Log in to Enterprise Manager Grid Control.

2. Navigate to the Application Servers tab.

3. Click **Add Application Servers**.

4. Select the Enterprise Manager Agent (must be 10.2.0.4 or higher) associated with the host where the application server instances are located.

Application server instances that are not collocated on the same computer as `ORACLE_HOME` must be registered before the instances can be discovered in Enterprise Manager Grid Control. The registry is a list of each application server instance's repository file. The repository files are generated when the application server instance is created. See Section 6.1.3 for more information on specifying a repository file for an application server instance. The following procedure is also required if a repository file location other than the default location is specified when creating an application server instance.

To discover distributed application server instances:

1. Create a file named `confighomeslst.add` and save it to the EM Agent's `/sysman/config/` directory on the host where the application server instances are located.

2. In the `confighomeslst.add` file, add the full path to each application server instance's repository file (`confighomes.lst`). For example:

    ```
    /shared/confighomes1/confighomes.lst
    /shared/confighomes2/confighomes.lst
    ```

If multiple users have created application server instances on the host, then the `confighomeslst.add` file will contain additional entries. If multiple instances use the same `confighomes.lst` file, than only a single entry is used and the application server instances are listed within the `confighomes.lst` file. For example:

```
# ConfigHome, <host>, OracleHome
/local/confighomes/user1/cfghome_1,  myHost1, /shared/oraclehomes/10131_PROD
/local/confighomes/user2/cfghome_1,  myHost2, /shared/oraclehomes/10131_PROD
```

> **Note:** For each host, `ORACLE_HOME` may be visible through a different mount point.

3. Save and close the file. Repeat this procedures on additional hosts if required.

4. Follow the previous Enterprise Manager Grid Control steps to discover the application server instances.

6.2 Adding and Deleting OC4J Instances

You can add or delete OC4J instances in an application server instance, as described in the following sections:

- Adding OC4J Instances
- Deleting OC4J Instances

6.2.1 Adding OC4J Instances

You can add OC4J instances to an application server instance in the following ways:

- With the `createinstance` script
- With Application Server Control

For example, you can add an additional OC4J instance to an Oracle WebCenter Framework installation. In Figure 6–2, a second OC4J instance named `crm` is added to the installation.

6.2.1.1 Adding OC4J Instances Using the Script

The `createinstance` script is used to create additional OC4J instances. The script is located in the `ORACLE_HOME/bin` directory.

> **Note:** As part of the creation process, you are asked to enter a password. This password will be tied to the `oc4jadmin` user for this instance. For consistency, you may want to enter the same password used to access the default OC4J instance with the `oc4jadmin` user.

The scripts usage is as follows:

```
createinstance -instanceName OC4J_instanceName [-configHome path] [-groupName
groupname] [-httpPort port] [-protocol protocol] [numProcs number]
[-startASControl] [-ASControlIsRoutable]
```

The arguments are described in Table 6–3.

Table 6–3   Arguments for createinstance Script

Argument	Description
-instanceName	A name for the OC4J instance to be created.
-configHome	Optional. The path of the application server instance to which the OC4J instance will be added. The OC4J instance is installed in ORACLE_HOME if no path is specified.
-groupName	Optional. The name of a group to which the OC4J instance will be allocated. The OC4J instance is allocated to the default_group group if no group name is specified.
-httpPort	Optional. The HTTP port used by the OC4J instance's default-web-site. The argument is only required if the OC4J instance will be used as a standalone J2EE Server. In this case, the OC4J instance uses the OC4J HTTP listener and does not use Oracle HTTP Server.
-protocol	Optional. The protocol used by the OC4J instance's default-web-site. The OC4J instance can use both the AJP and HTTP protocol. The HTTP protocol is used if no protocol is specified. Use the AJP protocol if the OC4J instance will be accessed by Oracle HTTP Server.
-numProcs	Optional. The number of processes that are started for this OC4J instance. One process is started if the number of processes is not specified.
-startASControl	Optional. Starts the Application Server Control when the OC4J instance is started.
-ASControlIsRoutable	Optional. Makes Application Server Control routable when the OC4J instance is started.

After creating an OC4J instance, start the new OC4J instance using OPMN. For example:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc process-type=oc4J_instanceName
(Windows) ORACLE_HOME\opmn\bin\opmnctl startproc process-type=oc4J_instanceName
```

If the OC4J instance is located on an application server instance other than the ORACLE_HOME instance, use the opmnctl script for the specific instance. For example:

```
(UNIX) ORACLE_CONFIG_HOME/opmn/bin/opmnctl startproc process-type=oc4J_
instanceName
(Windows) ORACLE_CONFIG_HOME\opmn\bin\opmnctl.bat startproc process-type=oc4J_
instanceName
```

6.2.1.2 Adding OC4J Instances Using Application Server Control

Application Server Control can be used to add additional OC4J instances to an application server instance.

To add an OC4J instance using Application Server Control:

1. Navigate to the Application Server *instance_name* page.

2. Click **Create OC4J Instance.**

3. On the Create OC4J Instance page, enter the following information:

   - **OC4J Instance Name:** Enter a name for the instance.

   - Select one of the following:

- **Add to an existing group with name:** Select a group from **Existing Group Name.**

- **Add to a new group with name:** In the **New Group Name field,** enter a name for the new group.

- Select **Start this OC4J instance after creation.**

4. Click **Create.**

   A confirmation screen is displayed after the instance has been created.

Note that the password used for this OC4J instance is the same password used for the `oc4jadmin` user for the installation.

Figure 6–2 shows part of the Cluster Topology page, which displays the additional OC4J instance added to the cluster.

Figure 6–2   OC4J Instance Added to Cluster

You can also verify that the instance was added by using the `opmnctl` command:

```
ORACLE_HOME\opmn\bin\opmnctl status
Processes in Instance: OracleAS_WC.sta.oracle.com
--------------------------------+--------------------+---------+---------
ias-component                   | process-type       |     pid | status
--------------------------------+--------------------+---------+---------
OC4JGroup:default_group         | OC4J:crm           |    9228 | Alive
OC4JGroup:default_group         | OC4J:OC4J_WebCent~ |    8616 | Alive
OC4JGroup:default_group         | OC4J:home          |    8615 | Alive
ASG                             | ASG                |     N/A | Down
```

Note: If you have configured Remote Management Interface (RMI) to use Secure Socket Layer (SSL), you must add the appropriate <ssl-config> element to the `rmi.xml` file for each OC4J instance you create. Otherwise, management connections to the OC4J instance from Application Server Control will either fail or use the non-secure RMI protocol, depending upon the value of the connection protocol property in the opmn.xml file for the administration OC4J instance. See Section A.3, "Configuring Security for Application Server Control" for more information.

See Also: The chapter, "Creating and Managing Additional OC4J Instances" in the *Oracle Containers for J2EE Configuration and Administration Guide*

6.2.2 Deleting OC4J Instances

You can delete an OC4J instance from an application server instance in the following
ways:

- With the `removeinstance` script

- With Application Server Control

Both methods delete the directory created for the OC4J instance and remove
configuration data for the instance from `opmn.xml`. Note the following guidelines for
deleting OC4J instances:

- You cannot delete the OC4J `home` instance that was created in `ORACLE_HOME`
  during installation.

- You can delete an OC4J `home` instance that is created in application server
  instances other than the `ORACLE_HOME` instance.

- You can delete OC4J instances that were created after installation.

6.2.2.1 Deleting OC4J Instance Using the Script

The `removeinstance` script is used to remove OC4J instances. The script is located
in the `ORACLE_HOME/bin` directory.

To delete an OC4J instance:

1. Stop the instance:

   ```
   (UNIX) ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=oc4J_instanceName
   (Windows) ORACLE_HOME\opmn\bin\opmnctl stopproc process-type=oc4J_instanceName
   ```

   If the OC4J instance is located on an application server instance other than the
   `ORACLE_HOME` instance, use the `opmnctl` script for the specific instance. For
   example:

   ```
   (UNIX) ORACLE_CONFIG_HOME/opmn/bin/opmnctl stopproc process-type=oc4J_
   instanceName
   (Windows) ORACLE_CONFIG_HOME\opmn\bin\opmnctl.bat stopproc process-type=oc4J_
   instanceName
   ```

2. Delete the instance:

   ```
   (UNIX) ORACLE_HOME/bin/removeinstance -instanceName oc4J_instanceName
   (Windows) ORACLE_HOME\bin\removeinstance.bat -instanceName oc4J_instanceName
   ```

   If the OC4J instance is located on an application server instance other than the
   `ORACLE_HOME` instance, specify the location of the application server instance
   using the `-configHome` argument. For example:

   ```
   (UNIX) ORACLE_HOME/bin/removeinstance -instanceName oc4J_instanceName
   -ConfigHome as_instance_location
   (Windows) ORACLE_HOME\bin\removeinstance.bat -instanceName oc4J_instanceName
   -ConfigHome as_instance_location
   ```

6.2.2.2 Deleting OC4J Instances Using Application Server Control

Application Server Control can be used to delete OC4J instances from an application
server instance.

To delete an OC4J instance with Application Server Control, take the following steps:

1. Navigate to the Application Server *instance_name* page.

2. Click the **Delete** icon for the instance you want to delete.

3. On the confirmation page, click **Yes.**

A confirmation screen is displayed after the instance has been deleted.

6.3 Configuring Cluster Topologies

A cluster topology is defined as two or more connected Oracle Application Server nodes.

Some reasons for creating a cluster include:

- Using a single instance of Application Server Control (the Administration OC4J instance) to manage all the instances in the cluster.

- Providing high availability of J2EE servers by installing multiple J2EE instances on multiple hosts. Oracle HTTP Server can route requests to the J2EE container, and the J2EE container can dynamically notify Oracle HTTP Server of new application bindings when an application is deployed. This scenario is described in Section 6.3.2.

- Perform some common administrative tasks automatically across multiple OC4J instances, by the use of groups. A **group** is a collection of OC4J instances that belong to the same cluster topology. Configuration operations can be executed simultaneously on all running OC4J instances in the group. See Section 2.3.3.2 for more information about groups and Section 6.3.3 for information about creating additional groups.

With this release, you can create the following types of cluster topologies:

- Dynamic node discovery: The cluster topology map for each node is automatically updated as nodes are added or removed, enabling the cluster to be self-managing.

- Static hubs as discovery servers: Specific nodes within a cluster are configured to serve as discovery servers, which maintain the topology map for the cluster; the remaining nodes then connect with one another through this server. Hubs in one topology can be connected to those in another.

- Connection of isolated topologies through gateways: This configuration is used to connect topologies that are separated by firewalls or on different subnets using specified "gateway" nodes.

- Manual node configuration: The host address and port for each node in the cluster are manually specified in the configuration. This is the same clustering mechanism supported in Oracle Application Server Release 2 (10.1.2) and is supported primarily to provide backward compatibility.

You can configure a cluster topology in the following ways:

- During installation, by checking the **Configure this instance to be part of an Oracle Application Server cluster** option in the Cluster Topology Configuration page. This method creates a dynamic node discovery cluster topology.

  See the *Oracle Application Server Installation Guide* for more information.

- After installation, by using Application Server Control:

  1. From the Cluster Topology page, click **Topology Network Configuration.**

  2. In the Topology section, select one of the following configurations:

  – **Configuring Dynamic Node Discovery Using Multicast:** For dynamic node discovery, enter the multicast address and port. For example:

```
225.0.0.33:8001
```

The multicast address must be in the range 224.0.1.0 - 239.255.255.255.

– **Configuring Static Discovery Servers:** For static discovery, enter the hostname or IP address and the OPMN remote port of the static discovery servers, separating the entries with commas.

– **Configuring Cross-Topology Gateways:** For each source node and target node, specify the host name or IP address of the server and its OPMN remote port. Separate the data for each node with an ampersand (&).

In addition, specify the multicast address and port, which will be used for dynamic discovery within the node's own cluster.

– **Configuring Static Node-to-Node Communication:** List the IP address and OPMN remote node of all the nodes you want to include in the cluster.

Figure 6–3 shows the Topology Network Configuration page:

Figure 6–3   Topology Network Configuration Page
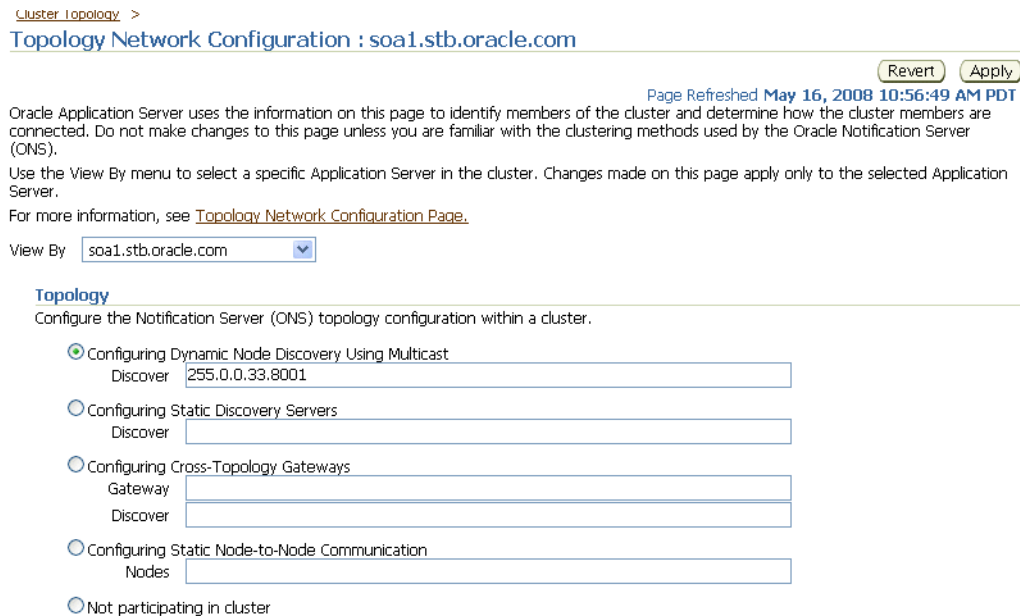
3. Click **Apply.**

■   After installation, by using one of the following Oracle Process Manager and Notification Server (OPMN) commands:

– `opmnctl`: This utility includes commands for updating opmn.xml with the multicast port:address and Web site configuration data needed to add an instance to a cluster. The syntax is:

```
opmnctl config topology update discover=*multicastAddress:multicastPort
```

– `opmnassociate`: This utility provides a one-step solution for adding an instance to a cluster. The syntax is:

```
opmnassociate *multicastAddress:multicastPort -restart
```

In the following sections, you create a cluster with three nodes, create two groups for OC4J instances, add OC4J instances to two nodes and add them to the groups, and specify multiple JVMs for the OC4J instances you create.

Figure 6–4 shows this configuration:

Figure 6–4   Multiple OC4J Middle Tiers, Additional OC4J Instances, and a Web Server Middle Tier in a Cluster
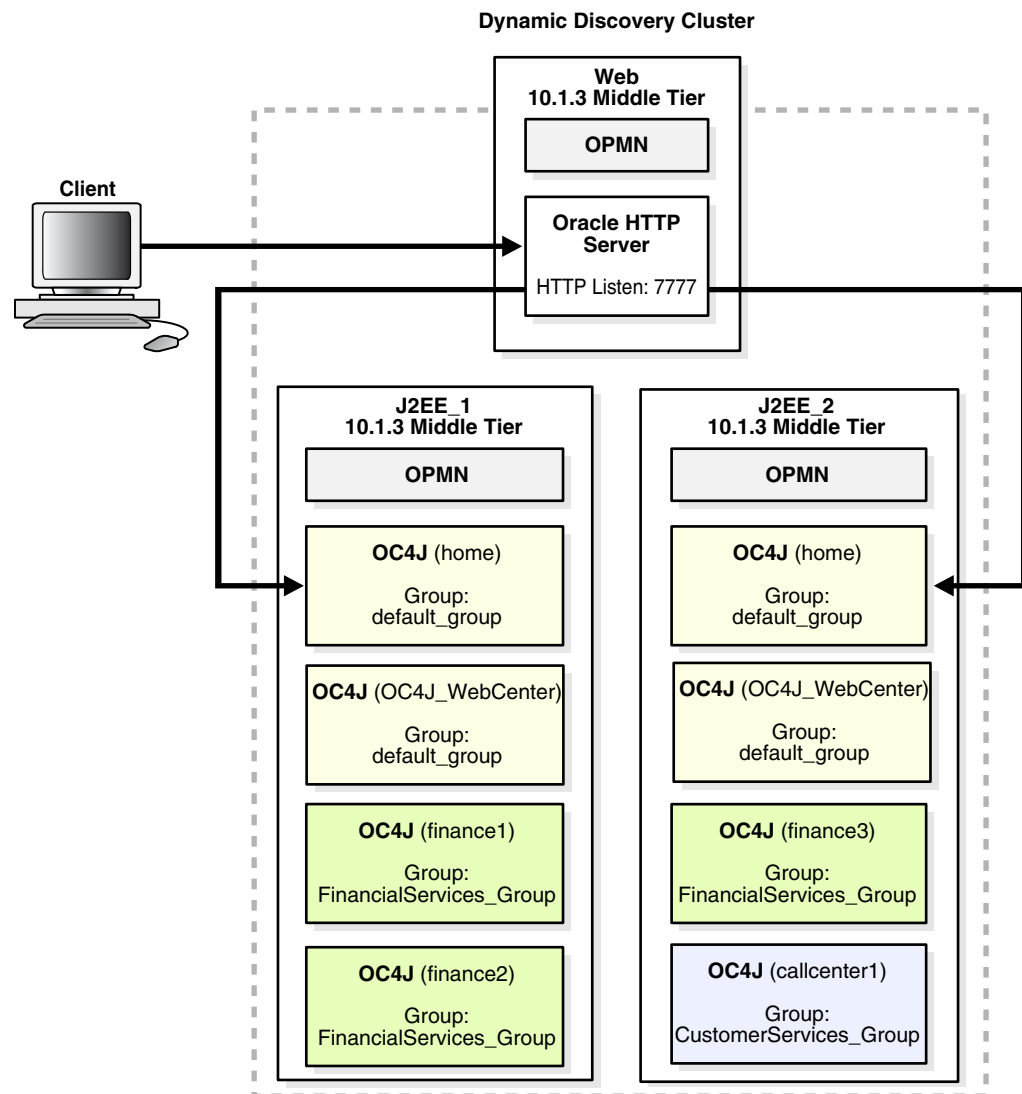
6.3.1 Configuring a Web Server and OC4J on Separate Hosts

In this example, you install a Web server, Oracle HTTP Server, on one host, and install Oracle WebCenter Framework, which includes Oracle Containers for J2EE (OC4J), on another host. Then, you cluster the instances using dynamic node discovery. This enables Oracle HTTP Server to route requests to the OC4J, and OC4J to dynamically

notify Oracle HTTP Server of new application bindings when an application is deployed.

Figure 6–5 shows this environment.

Figure 6–5 Web Server Middle Tier and Oracle WebCenter Framework Middle Tier on Separate Hosts in a Cluster
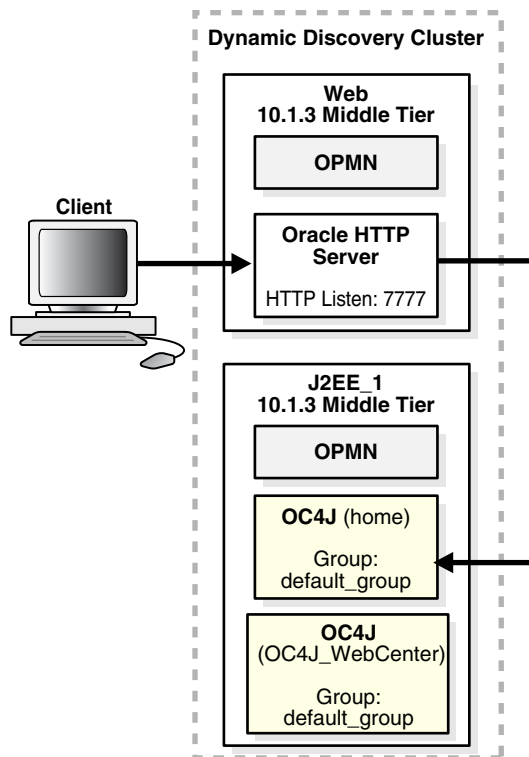
For this scenario, install the following Oracle Application Server middle-tier instances on separate hosts. (For this scenario, do not configure the cluster during installation.)

- Oracle HTTP Server (referred to in this procedure as Web), which includes Oracle HTTP Server and OPMN

    When you install Oracle HTTP Server, accept the defaults on the Cluster Topology Configuration page.

- Oracle WebCenter Framework only (referred to in this procedure as J2EE_1), which includes OC4J, Oracle WebCenter Framework, and OPMN

    When you install Oracle WebCenter Framework, on the Administration Settings page, select **Start Oracle Enterprise Manager 10g ASControl in this instance.** This sets the OC4J instance to be the Administration OC4J instance.

    On the Cluster Topology Configuration page, accept the defaults. You will add the instances to the cluster later.

    Note the following points:

    – When you select an OC4J instance to be the Administration OC4J instance, the Application Server Control that is hosted within that instance administers the local OC4J instance and any other instances in the cluster that are not designated as Administration OC4J instances.

If you did not choose this option, you can configure it later, by starting the ascontrol application in the instance:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc application=ascontrol
(Windows) ORACLE_HOME\opmn\bin\opmnctl startproc application=ascontrol
```

- For those instances that are not specified as the Administration OC4J instance, Application Server Control is deployed in the instance, but not started.

- Oracle recommends that only one OC4J instance in the cluster be configured as the Administration OC4J instance.

After you install the middle-tier instances, configure the instances for dynamic node discovery, by taking the following steps:

1. To configure the Oracle HTTP Server instance for dynamic node discovery, set the OPMN multicast discovery address using the `opmnctl config topology` command. (You cannot use Application Server Control because it is not running on this instance; you cannot use `opmnassociate` because you used a name other than `home` for the default OC4J instance.)

   For example, to associate the Oracle HTTP Server instance on UNIX to the multicast address 225.0.0.33, use the following commands:

   ```
   ORACLE_HOME/opmn/bin/opmnctl config topology update discover=*225.0.0.33:8001
   ORACLE_HOME/opmn/bin/opmnctl reload
   ```

2. To configure the Oracle WebCenter Framework instance for dynamic node discovery, you can use the `opmnctl config topology` command or Application Server Control. For this example, use Application Server Control and take the following steps:

   a. From the Cluster Topology page of Application Server Control, click **Topology Network Configuration.**

   b. In the Topology section, select **Configuring Dynamic Node Discovery Using Multicast**. Then, enter the same multicast address and port that you used for the Oracle HTTP Server instance. For example:

   ```
   225.0.0.33:8001
   ```

   c. Click **Apply.**

   Now, both instances are part of the same cluster topology.

Verify the configuration in one of the following ways:

- Using Application Server Control: Navigate to the Cluster Topology page. The page displays both instances, as shown in Figure 6–6:

Figure 6–6   Verifying the Cluster Topology

- Using the `opmnctl` command with the `@cluster` option. The following example shows the output:

```
ORACLE_HOME/opmn/bin/opmnctl @cluster status
Processes in Instance: J2EE_1.sta.oracle.com
-------------------------------+-------------------+---------+---------
ias-component                  | process-type      |     pid | status
-------------------------------+-------------------+---------+---------
OC4JGroup:default_group        | OC4J:OC4J_WebCent~|    8616 | Alive
OC4JGroup:default_group        | OC4J:home         |    8615 | Alive
ASG                            | ASG               |     N/A | Down

Processes in Instance: Web.stad.oracle.com
-------------------------------+-------------------+---------+---------
ias-component                  | process-type      |     pid | status
-------------------------------+-------------------+---------+---------
HTTP_Server                    | HTTP_Server       |   25118 | Alive
```

6.3.2 Configuring Multiple J2EE Server Middle Tiers in a Cluster

The example in this section builds upon the example in the previous section, Section 6.3.1. It adds an additional J2EE server middle tier (Oracle WebCenter Framework installation) to the cluster topology, to support a highly available environment for testing or production purposes.

Figure 6–7 shows this environment.

Figure 6–7   Multiple J2EE Server Middle Tiers and a Web Server Middle Tier in a Cluster
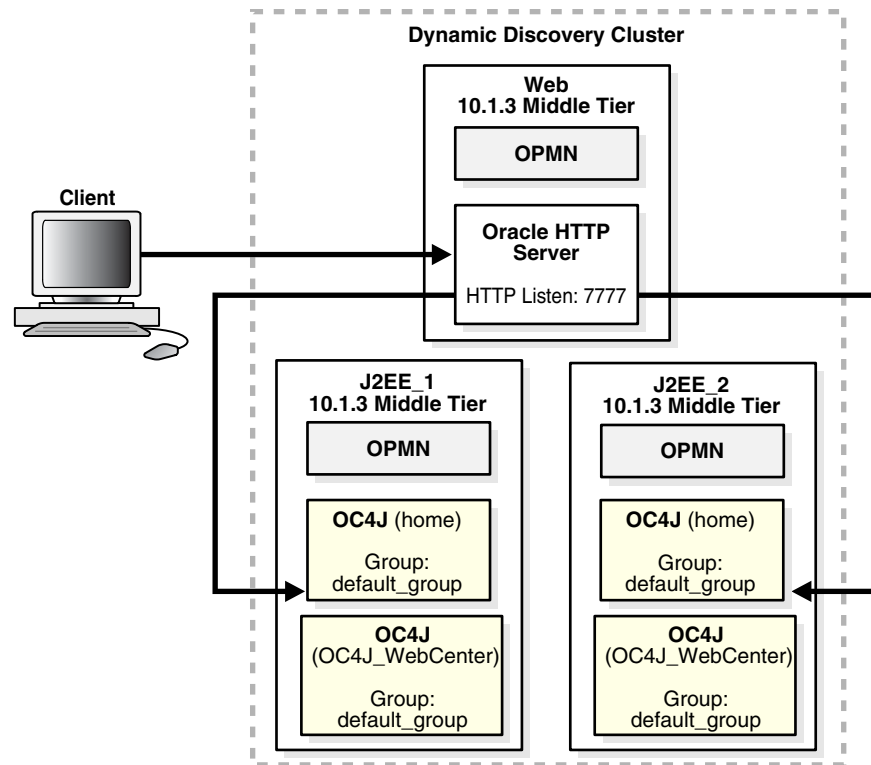
For this scenario:

- Install and configure instances as described in Section 6.3.1.

- Install an additional Oracle WebCenter Framework instance (referred to in this procedure as J2EE_2).

    During installation, do not select **Start Oracle Enterprise Manager 10g ASControl in this instance.** in the Administration Settings page. This instance will be managed by the Administration OC4J instance in J2EE_1 after you add it to the cluster.

Then, configure this instance for dynamic node discovery, adding it to the same cluster as the previous instances, using the opmnctl command.

For example, to associate the J2EE_2 instance on UNIX to the multicast address 225.0.0.33, use the following command:

```
ORACLE_HOME/opmn/bin/opmnctl config topology update discover=*225.0.0.33:8001
ORACLE_HOME/opmn/bin/opmnctl reload
```

Now, this instance is part of the cluster topology and is managed by the Administration OC4J instance in J2EE_1. This OC4J instance uses Apache JServ Protocol (AJP) to receive and respond to requests from Oracle HTTP Server.

Verify the configuration using Application Server Control or the opmnctl @cluster status command. For example, to verify it using Application Server Control, navigate to the Cluster Topology page. The page displays all three instances in the Members section, as shown in Figure 6–8.

Figure 6–8 Verifying the Updated Cluster Topology

The default OC4J instance, `home`, and the `OC4J_WebCenter` instance are part of the group `default_group`. Figure 6–9 shows the Groups section of the Cluster Topology page.

Figure 6–9 The default_group Group

6.3.3 Creating Additional Groups

A **group** is a collection of OC4J instances that belong to the same cluster topology. Configuration operations can be executed simultaneously on all running OC4J instances in the group.

You can create additional groups. For this scenario, create two empty groups:

- FinancialServices_Group
- CustomerServices_Group

Take the following steps for each group:

1. From the Groups section of Cluster Topology page, click **Create.**

2. For **Group Name,** enter `FinancialServices_Group`.

3. Click **Create.**

4. Repeat steps 1 through 3, entering `CustomerServices_Group` as the **Group Name.**

In the next section, you will create new OC4J instances and add them to the groups.

6.3.4 Adding OC4J Instances and Adding Them to Groups

As described in Section 6.2, you can add OC4J instances to an existing Oracle home. In this section, you add the instances and add them to the groups created in Section 6.3.3, as shown in the following table:

Application Server Instance	OC4J Instance Name	Group Name
J2EE_1	finance1	FinancialServices_Group
J2EE_1	finance2	FinancialServices_Group
J2EE_2	finance3	FinancialServices_Group
J2EE_2	callcenter1	CustomerServices_Group

Take the following steps for each of the four OC4J instances you create, using the information in the preceding table:

1. Navigate to the Application Server *instance_name* page, for example, Application Server J2EE_1.*hostname*.

2. Click **Create OC4J Instance.**

3. On the Create OC4J Instance page, enter the following information:

   - **OC4J Instance Name:** Enter a name for the instance. For example, for the J2EE_1 instance, enter `finance1`.

   - Select **Add to an existing group with name**, then select the appropriate group from **Existing Group Name.**

4. Select **Start this OC4J instance after creation.**

5. Click **Create.**

   A confirmation screen is displayed after the instance has been created.

Figure 6–10 shows the Members section of the Cluster Topology page, with the new OC4J instances displayed.

Figure 6–10   New OC4J Instances Displayed in Cluster Topology Page

Figure 6–11 shows the Groups section of the Cluster Topology page, with the new groups and their members.

Figure 6–11   New Groups Displayed in Cluster Topology Page

Now, the cluster is configured as depicted in Figure 6–4 in Section 6.3.

6.3.5 Creating Multiple JVMs

OC4J executes on the Java Virtual Machine (JVM) of the standard Java Development Kit (JDK). By default, each OC4J instance uses one JVM. However, you can configure an OC4J instance so it runs on multiple JVMs.

In this case, the OC4J instance is essentially running on multiple processes. This can improve performance and provide a level of fault tolerance for your deployed applications. However, multiple JVMs also require additional hardware resources to run efficiently.

> **Note:** You cannot configure the OC4J instance that is hosting the active Application Server Control (represented by the `ascontrol` application) to run multiple JVMs.

In this example, you create an additional JVM for each OC4J instance you created in Section 6.3.4. Take the following steps for each of these OC4J instances:

1. Navigate to the Home page for the OC4J instance.

2. Click **Administration.**

3. If necessary, expand the Properties section of the table by clicking the Expand icon. Then, click the **Go to Task** icon in the **Server Properties** row.

4. In the **Number of VM Processes** field, enter the number of JVMs to configure.

5. Click **Apply.**

6. Navigate to the Cluster Topology page, select the OC4J instance that you modified, and click **Restart.** On the Confirmation page, click **Yes.**

6.4 Configuring 10.1.2 OracleAS Web Cache as a Reverse Proxy

You can use Release 2 (10.1.2) OracleAS Web Cache as a reverse proxy for your 10*g* Release 3 (10.1.3.4) middle-tier instance. As a reverse proxy server, OracleAS Web Cache acts a gateway to the middle-tier servers.

The following topics describe how to configure OracleAS Web Cache Release 2 (10.1.2) as a reverse proxy for your 10*g* Release 3 (10.1.3.4) middle-tier instance:

- Configuring an OracleAS Web Cache Instance as a Reverse Proxy
- Configuring an OracleAS Web Cache Cluster as a Reverse Proxy

> **See Also:** For complete information about using OracleAS Web Cache as a reverse proxy, and about OracleAS Web Cache clusters, see the Release 2 (10.1.2) *Oracle Application Server Web Cache Administrator's Guide*.

6.4.1 Configuring an OracleAS Web Cache Instance as a Reverse Proxy

You can use Release 2 (10.1.2) OracleAS Web Cache as a reverse proxy for your middle-tier instance. The procedure in this section makes the following assumptions:

- You have installed a Release 2 (10.1.2) OracleAS Web Cache standalone kit. The kit is part of the Oracle Application Server Companion CD, which is available on OTN.
- You have configured OracleAS Web Cache, as described in the *Oracle Application Server Web Cache Administrator's Guide*.
- You have installed a 10*g* Release 3 (10.1.3.4) middle-tier instance.

Figure 6–12 depicts the scenario described in this section.

Figure 6–12   OracleAS Web Cache as Reverse Proxy
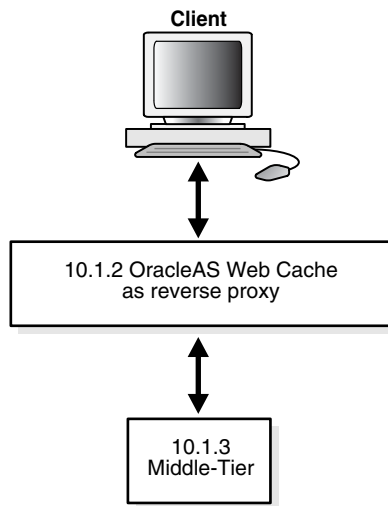
From the Release 2 (10.1.2) standalone OracleAS Web Cache instance, take the following steps:

1. Log into OracleAS Web Cache Manager, using the OracleAS Web Cache username and password. By default, the username is `ias_admin` and the password is the one you specified during installation. Use the following URL, where *port* is the OracleAS Web Cache administration port:

   `http://hostname:port/webcacheadmin`

   By default, the port is 9400. The administration port number for a OracleAS Web Cache standalone installation is listed in the `Oracle_Home`/webcache/webcache.xml file. To find the port number for OracleAS Web Cache that is part of an Oracle Application Server installation, click the **Ports** link in Application Server Control.

2. In the navigator frame, select **Origin Servers, Sites, and Load Balancing** > **Origin Servers.**

3. In the Origin Servers page, click **Add** in the Application Web Servers section.

4. In the Add Application Web Server dialog box, enter the following information:

   - In the **Hostname** field, enter the host name of the origin server (Oracle HTTP Server) in the 10*g* Release 3 (10.1.3.4) middle-tier instance.

   - In the **Port** field, enter the listening port from which the origin server will receive OracleAS Web Cache requests.

   - In the **Routing** field, select ENABLED to permit OracleAS Web Cache to route requests to the origin server.

   For information about other fields in the dialog box, refer to the online Help or the *Oracle Application Server Web Cache Administrator's Guide*.

5. Click **Submit.**

6. Optionally, you can add a new site to map to the origin server or you can use an existing site. To add a new site, in the navigator frame, select **Origin Servers, Sites, and Load Balancing** > **Site Definitions**.

   For more information about adding a site, refer to the online Help or the *Oracle Application Server Web Cache Administrator's Guide*.

7. In the navigator frame, select **Origin Servers, Sites, and Load Balancing** > **Site-to-Server Mapping** to map the site to the origin server in the 10*g* Release 3 (10.1.3.4) middle-tier instance.

8. In the Site-to-Server Mapping page, select a mapping and click **Insert Above** or **Insert Below**.

9. In the Edit/Add Site-to-Server Mapping dialog box:

   - Select **Select from Site definitions** to select the site definition you want to use.

   - In the **Select Application Web Servers** field, select the application Web server from 10*g* Release 3 (10.1.3.4) middle-tier instance.

6.4.2 Configuring an OracleAS Web Cache Cluster as a Reverse Proxy

You can use a cluster of Release 2 (10.1.2) OracleAS Web Cache instances as a reverse proxy for your 10*g* Release 3 (10.1.3.4) middle-tier instance.

The procedure in this section makes the following assumptions:

- You have installed more than one instance of Release 2 (10.1.2) OracleAS Web Cache standalone kit. The kit is part of the Oracle Application Server Companion CD, which is available on OTN.

- You have configured OracleAS Web Cache, as described in the *Oracle Application Server Web Cache Administrator's Guide*.

- You have installed a 10*g* Release 3 (10.1.3.4) middle-tier instance.

Figure 6–13 depicts the scenario described in this section.

Figure 6–13   OracleAS Web Cache Cluster as Reverse Proxy
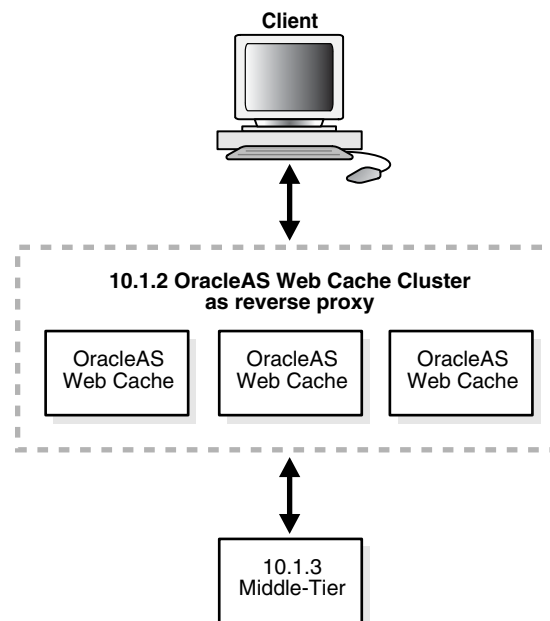
To configure an OracleAS Web Cache cluster as a reverse proxy, take the following steps:

1. Set up one OracleAS Web Cache instance as a reverse proxy, as described in Section 6.4.1.

2. Log into OracleAS Web Cache Manager for that instance, using the OracleAS Web Cache username and password. By default, the username is `ias_admin` and the password is the one you specified during installation. Use the following URL, where *port* is the OracleAS Web Cache administration port:

   ```
   http://hostname:port/webcacheadmin
   ```

3. Configure the properties of the cache cluster, by taking the following steps:

   a. In the OracleAS Web Cache Manager navigator frame, select **Properties** > **Clustering.**

   b. In the **General Cluster Information** section of the Clustering page, click **Edit.**

   Follow the directions in the online Help or Chapter 10 of the *Oracle Application Server Web Cache Administrator's Guide*.

4. Add other caches to the cluster, by taking the following steps for each instance you want to place in the cluster:

   a. In the OracleAS Web Cache Manager navigator frame, select **Properties** > **Clustering.**

   b. In the **Cluster Members** section of the Clustering page, click **Add.**

   Follow the directions in the online Help or Chapter 10 of the *Oracle Application Server Web Cache Administrator's Guide*.

5. After you have added all the caches to the cluster, propagate the cluster configuration to the cluster members by taking the following steps:

   a. In the OracleAS Web Cache Manager navigator frame, select **Operations** > **Cache Operations.**

   b. Select **All Caches,** and click **Propagate.**

   c. Restart all caches by selecting **All Caches** and clicking **Restart.**

6.5 Configuring Oracle Application Server 10.1.2 with Oracle Application Server 10.1.3

To use the latest J2EE features of Oracle Application Server, 10*g* Release 3 (10.1.3.4), with existing Oracle Application Server, Release 2 (10.1.2), components and applications, you can use the Oracle HTTP Server from an Oracle Application Server, Release 2 (10.1.2), middle tier as the front-end for your Oracle Application Server, 10*g* Release 3 (10.1.3.4), middle tier. This section provides instructions on installing and configuring Oracle Application Server, 10*g* Release 3 (10.1.3.4), and Oracle HTTP Server, Release 2 (10.1.2), for interoperability.

On your farm or cluster, either install or locate the following:

- **Server 1**—An Oracle Application Server, Release 2 (10.1.2), J2EE and Web Cache type middle tier or other Oracle Application Server, Release 2 (10.1.2), middle tier with Oracle HTTP Server.

- **Server 2**—An Oracle Application Server, 10*g* Release 3 (10.1.3.4), Oracle WebCenter Framework middle tier. For instructions on installing Oracle Application Server, 10*g* Release 3 (10.1.3.4) refer to the 10*g* Release 3 (10.1.3.4), *Oracle Application Server Installation Guide* for your platform. This instance must use AJP protocol. If you are using HTTP protocol, run the following commands to change to AJP protocol:

   ```
   ORACLE_HOME_SERVER2/opmn/bin/opmnctl config port update ias-component=default_
   ```

```
group process-type=instance name portid=default-web-site protocol=ajp
range=12501-12600
ORACLE_HOME_SERVER2/opmn/bin/opmnctl reload
ORACLE_HOME_SERVER2/opmn/bin/opmnctl restartproc ias-component=default_group
process-type=instance name
```

For Release 2 (10.1.2) middle tiers that are associated with an Infrastructure, start with Step 3 in the following procedure because the ons.conf file on the Release 2 (10.1.2) middle tier is not updated in this configuration. For this configuration, the Oracle Application Server, 10*g* Release 3 (10.1.3.4), initiates the connections between the two instances. For middle tiers that are not associated with an Infrastructure (a J2EE and Web Cache type middle tier may or may not be associated with an Infrastructure), start with Step 1.

Configuration Procedure

Perform the following steps to configure the two servers:

1. On Server 1, add Server 2 using DCM:

   ```
   ORACLE_HOME/dcm/bin/dcmctl addOPMNLink server2_ip:server2_ons_remote_port
   ```

   In the example:

   - *server2_ip* is the IP address of Server 2. To find the IP address, you can use the ping command:

     ```
     ping server_name
     ```

   - *server2_ons_remote_port* is the remote ONS port for the server. The port number is located in the opmn.xml file. In the following example, the remote port is 6200.

     ```
     <notification-server interface="ipv4">
             <port local="6100" remote="6200" request="6003"/>
     ```

2. Verify that Server 2 is added by inspecting the contents of ons.conf in the *ORACLE_HOME*/opmn/conf directory. The file should contain a list of comma-separated *hostname/ip:ons_remote_port* entries. The remote port is the port on Server 2 that OPMN on Server 1 uses to communicate with Server 2. An example of an entry in the list would be:

   ```
   127.2.148.142:6200
   ```

3. On Server 2, add Server 1 to the cluster using static node-to-node communication by editing the topology section in *ORACLE_HOME*/opmn/conf/opmn.xml:

   ```
   <notification-server>
   ...
      <topology>
         <nodes list="server1_ip:remote_port,server2_ip:remote_port"/>
      </topology>
   </notification-server>
   ```

   In the example, *server*_ip* is the IP address of Server 1 or Server 2 and *remote_port* is the port number through which other servers can communicate with the server, for example:

   ```
   127.2.148.142:6200
   ```

To configure ONS in opmn.xml, all Oracle RAC instance nodes must be listed, not only the master. For an Oracle RAC with host1, host2, host3, host4, the list would be:

```
list="host1:ONSRemotePort,host2:ONSRemotePort,host3:ONSRemotePort,host4:ONSRemo
tePort"
```

In an Oracle RAC environment, the SSL setting must be the same on all nodes: enabled or disabled.

4. On Server 2, reload OPMN:

    ```
    ORACLE_HOME_SERVER2/opmn/bin/opmnctl reload
    ```

5. Verify that both servers can communicate with each other by running the following commands:

    ■   On Server 1:

        ```
        ORACLE_HOME_SERVER1/opmn/bin/opmnctl @farm status
        ```

    ■   On Server 2:

        ```
        ORACLE_HOME_SERVER2/opmn/bin/opmnctl @cluster status
        ```

        These commands produce a list of servers that are part of the farm or cluster.

6. On Server 1, set the OC4J mount directive in the ORACLE_ HOME/Apache/Apache/conf/mod_oc4j.conf file:

    ```
    Oc4jMount /MyApp instance://server2_instance_name:oc4j_instance_name
    Oc4jMount /MyApp/* instance://server2_instance_name:oc4j_instance_name
    ```

    A mount point must be added for each J2EE application deployed on 10*g* Release 3 (10.1.3.4). As new applications are added, you must add a new mount point.

7. On Server 1, update the configuration. (You must update the configuration any time you manually edit mod_oc4J.conf.)

    ```
    ORACLE_HOME/dcm/bin/dcmctl updateConfig
    ```

    > **See Also:** *Distributed Configuration Management Administrator's Guide*

8. If the Oracle Application Server 10*g* Release 3 (10.1.3.4) instance contains Oracle Content DB, you must change Oracle Content DB properties to refer to the 10.1.2.0.2 Oracle HTTP Server host name and port number.

    You change the properties using Application Server Control:

    a. Navigate to the OC4J_Content home page, and select **Applications.**

    b. Click **content,** then click **Content DB Extension.**

    c. Select the Administration tab.

    d. In the **Domain Properties** row, click the **Go to Task** icon.

    e. Click **IFS.DOMAIN.APPLICATION.ApplicationHost.** Change the host name in the **Value** field. Then, click **OK.**

    f. Click **IFS.DOMAIN.APPLICATION.ApplicationPort.** Change the port number in the **Value** field. Then, click **OK.**

9. If the Oracle Application Server 10*g* Release 3 (10.1.3.4) instance uses OracleAS Single Sign-On, take the following steps:

a. On Server 1, perform Steps 1 and 2 in "Task 1: Enable SSO Authentication (Optional)" in Section 6.7.

b. Copy the newly created osso configuration file to the following location in the Server 2, 10*g* Release 3 (10.1.3.4), instance:

```
(UNIX) ORACLE_HOME/Apache/Apache/conf/osso
(Windows) ORACLE_HOME\Apache\Apache\conf\osso
```

c. In the httpd.conf file, remove the comment character (#) from the line that includes `mod_osso.conf`.

d. In the mod_oc4j.conf file, add an entry for the osso configuration file:

```
OssoConfig new_osso.conf_file_path
```

e. On Server 1, update the configuration:

```
ORACLE_HOME/dcm/bin/dcmctl updateConfig
```

10. Associate Server 2 with Oracle Internet Directory, as described in "Task 2: Configure the Middle-Tier Instance for Identity Management" of Section 6.7.

11. Restart Oracle HTTP Server on Server 1:

```
ORACLE_HOME_SERVER1/opmn/bin/opmnctl restartproc ias-component=HTTP_Server
```

A browser pointing to `appserverInstance` on Server 1 can now access an OC4J application in `appserverInstance` on Server 2.

> **Caution:** For Release 2 (10.1.2) middle-tier installations other than a J2EE and Web Cache type middle tier, start the Oracle Application Server, Release 2 (10.1.2), middle tier before starting the Oracle Application Server, 10*g* Release 3 (10.1.3.4), middle tier, or a delay of up to two minutes can occur.

In this configuration, the application stop functionality in the J2EE 10*g* Release 3 (10.1.3.4) instance should not be used because errors can occur when the Oracle HTTP Server Release 2 (10.1.2) routes to the J2EE 10*g* Release 3 (10.1.3.4) instance with the stopped application.

> **Note:** You cannot manage Release 2 (10.1.2) from the 10*g* Release 3 (10.1.3.4) Application Server Control. The 10.1.2 instance will appear on the Cluster Topology page for Application Server Control, 10*g* Release 3 (10.1.3.4), but some of the information about the 10.1.2 instance will be either unavailable or incorrect. For example:
>
> ■ In the Ports page, the ports for the 10.1.2 instance may not be listed or the type of port may be listed as NA.
>
> ■ In the Groups section of the topology page, if a group contains a 10.1.2 instance, the status of the instance may be incorrect.

6.6 Configuring Instances to Use OC4J Java Single Sign-On

You can configure instances to use OC4J Java Single Sign-On (Java SSO), a lighter-weight single sign-on solution supplied with OC4J that does not require additional infrastructure (such as OracleAS Single Sign-On and Oracle Access

Manager single sign-on do) and decouples OC4J from any identity management system that you use.

Note that for a Basic Installation, Java SSO is automatically deployed, configured, and started. For an Advanced Installation, it is deployed, but not configured or started.

For information about configuring instances to use Java SSO, see the section, "Java SSO Setup and Configuration" in the *Oracle Containers for J2EE Security Guide*.

For information about configuring Java SSO to use Oracle Internet Directory, see the section "Associate Oracle Internet Directory with OC4J" in the *Oracle Containers for J2EE Security Guide*.

For information about configuring Oracle WebCenter Framework to use Java SSO, see the section "Configuring a WebCenter Application to Use Java Single Sign-On" in the *Oracle WebCenter Framework Developer's Guide*.

6.7 Configuring Instances to Use 10.1.4 or 10.1.2 Oracle Identity Management

You can configure 10.1.3 middle-tier instances to use Release 10.1.4 or Release 2 (10.1.2) Oracle Identity Management.

This section describes how to configure a 10.1.3 middle-tier instance to use Oracle Identity Management, Release 10.1.4 or Release 2 (10.1.2). Figure 6–14 shows the middle-tier instance configured with Oracle Identity Management, Release 2 (10.1.2).

Figure 6–14   Middle Tier Using 10.1.2 Identity Management
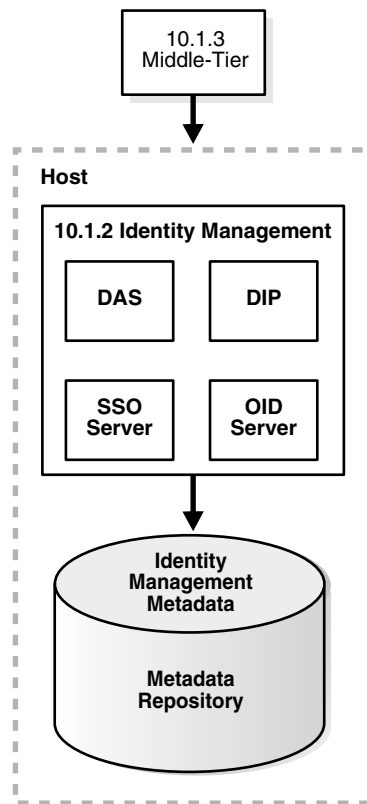
Before you start, make sure that:

■ The Oracle Identity Management instance is started (status is Up).

■ You know the Oracle Internet Directory host and port numbers.

■ You know the password for `cn=orcladmin`, or another user who is a member of the `iASAdmins` group.

■ For information about configuring Oracle WebCenter Framework to use Oracle Internet Directory and OracleAS Single Sign-On, see the section "Configuring a WebCenter Application to use LDAP and Single Sign-On" in the *Oracle WebCenter Framework Developer's Guide*.

Task 1: Enable SSO Authentication (Optional)

If you want to enable OracleAS Single Sign-On authentication for deployed applications, you must take the following steps before you use the Identity Management wizard described in "Task 2: Configure the Middle-Tier Instance for Identity Management".

> **Note:** OracleAS Single Sign-On is available only if you have installed an Oracle Application Server installation type that includes Oracle HTTP Server. For other installation types, Java SSO is available. See Section 6.6.

1. On the Identity Management host, set the environment variables ORACLE_HOME and ORACLE_SID.

2. On the Identity Management host, run the `ssoreg` script, using the `-remote_midtier` option. The file is located at:

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

   For example, on LINUX:

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

   The resulting configuration file (`mysso.conf` in the example) is an obfuscated osso configuration file.

3. Copy the obfuscated osso configuration file to the 10*g* Release 3 (10.1.3.4) middle-tier instance.

4. On the middle-tier host, run the following script to complete the registration:

   ```
   (UNIX) ORACLE_HOME/Apache/Apache/bin/osso1013 config_file
   (Windows) perl ORACLE_HOME\Apache\Apache\bin\osso1013 config_file
   ```

Task 2: Configure the Middle-Tier Instance for Identity Management

To configure the middle-tier instance to use Identity Management, take the following steps:

1. Using Application Server Control, navigate to the OC4J Home page for the middle-tier instance.

2. Click **Administration.**

3. In the Task Name column of the table, expand **Security** if it is not already expanded. Then, in the **Identity Management** row, click the **Go to Task** icon.

4. On the Identity Management page, click **Configure.**

5. On the Configure Identity Management: Connect Information page:

   - **Oracle Internet Directory Host:** Enter the fully qualified name of the Oracle Internet Directory host.

   - **Oracle Internet Directory User DN:** Enter the distinguished name of a user, such as cn=orcladmin, in the iASAdmins group.

   - **Password:** Enter the password for the user.

     This password will be used as the default password for the oc4jadmin user created in Oracle Internet Directory.

   - **Use only SSL connections to the Internet Directory:** Select this option if you want middle-tier components to connect to Oracle Internet Directory using only SSL.

     Then, in the **Oracle Internet Directory SSL Port** field, enter the Oracle Internet Directory SSL port number.

   - **Use non-SSL connections to the Internet Directory:** Select this option if you want middle-tier components to connect to Oracle Internet Directory using non-SSL connections.

     Then, in the **Oracle Internet Directory Port** field, enter the Oracle Internet Directory non-SSL port number.

   Click **Next.**

6. On the Configure Identity Management: Application Server Control page, you can specify if you want to configure Application Server Control to use Identity Management as its security provider, for authentication and authorization of administrative users. If you do, select **Use Oracle Identity Management Security Provider.**

   Note the following:

   – Any Application Server Control administrator users created for the current security provider will not be able to access Application Server Control after you make this change. Only users and groups defined in Oracle Internet Directory will be able to access Application Server Control.

   – You can change the Application Server Control security provider later by clicking **Setup**, then **Security Provider.**

7. On the Configure Identity Management: Deployed Applications page, you can specify security options for applications deployed in this OC4J instance. For each application:

   - **Use OID Security Provider:** Select this option to configure applications to use Identity Management as its security provider for authentication and authorization.

     Note that you cannot change the security provider for the default application.

   - **Enable SSO Authentication:** If you selected **Use OID Security Provider**, you can select this option to use Single Sign-On authentication. Note, however, that you must have first registered your instance of Oracle Application Server with the OracleAS Single Sign-On server. See "Task 1: Enable SSO Authentication (Optional)" for more information.

Click **Configure.**

8. When the operation completes, you must restart the OC4J instance. Do not click Restart on the Confirmation page. Instead, navigate to the Cluster Topology page, select the OC4J instance, and click **Restart.**

Your middle tier is now configured to use Oracle Identity Management services.

> **See Also:** *Oracle Identity Management Concepts and Deployment Planning Guide*, Release 2 (10.1.2)

6.8 Disabling and Enabling Anonymous Binds

Beginning with Release 2 (10.1.2.0.2), you can enable and disable anonymous binds (anonymous authentication) in Oracle Internet Directory. By default, anonymous binds are enabled.

Although disabling anonymous binds is useful in many run-time environments, most configuration changes, such as the following, require that anonymous binds are enabled:

- Installing new components with Oracle Universal Installer
- Configuring components with Application Server Control
- Changing the host name, domain name, or IP address of a host on which you have installed Oracle Application Server

6.8.1 Disabling Anonymous Binds for Run-Time Environments

To disable anonymous binds, take the following steps:

1. Shut down all middle tiers that are connected to the OracleAS Infrastructure, as described in Section 3.2.2.

2. Shut down OracleAS Infrastructure, in all Infrastructure Oracle homes:

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

3. Start Oracle Internet Directory, because it must be started while you perform the procedure:

```
(UNIX) ORACLE_HOME/bin/oidmon connect=db_connect_string start
(Windows) ORACLE_HOME\bin\oidmon connect=db_connect_string start
```

4. Edit the `ias.properties` file for each middle tier connected to the OracleAS Infrastructure and for the Infrastructure Oracle home that contains OracleAS Single Sign-On and Oracle Delegated Administration Services. The `ias.properties` file is located in the following directory:

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

In the `ias.properties` file, add the `OIDAnonymousDisabled` property to the file and set it to `true`:

```
OIDAnonymousDisabled=true
```

5. Edit the `dads.conf` file for each middle tier connected to the OracleAS Infrastructure and for the Infrastructure Oracle home that contains OracleAS

Single Sign-On and Oracle Delegated Administration Services. The `dads.conf` file is located in the following directory:

```
(UNIX) ORACLE_HOME/Apache/modplsql/conf
(Windows) ORACLE_HOME\Apache\modplsql\conf
```

By default, the PlsqlDatabaseConnectString parameter contains a value that uses the LDAP name resolution format, for example:

```
PlsqlDatabaseConnectString cn=orcl, cn=oraclecontext NetServiceNameFormat
```

Comment out this line. (Do not delete it because you will need to revert to it if you want to enable anonymous binds in the future.)

Add the following line, which changes the value of the PlsqlDatabaseConnectString parameter to use the `host:port:service` format instead of LDAP name resolution:

```
PlsqlDatabaseConnectString db_host:db_hostdb_listener_port:db_service_name
```

In the example, *db_host* is the name of the host on which the OracleAS Metadata Repository for OracleAS Single Sign-On is installed, *db_listener_port* is the listener port for that OracleAS Metadata Repository, and *db_service_name* is the service name for the OracleAS Metadata Repository.

6. Use the `ldapmodify` command to disable anonymous binds. Use the command on the Oracle home that contains Oracle Internet Directory.

   Take the following steps:

   a. Create a text file with the following lines:

      ```
      dn:
      changetype: modify
      replace: orclanonymousbindsflag
      orclanonymousbindsflag: 0
      ```

   b. Use the `ldapmodify` command, calling the text file created in the previous step as input. In the following example, the text file is named `anon_off.ldif`:

      ```
      (UNIX) ORACLE_HOME/bin/ldapmodify -h host -p port -D cn=orcladmin -w
      password -v -f anon_off.ldif
      (Windows) ORACLE_HOME\bin\ldapmodify -h host -p port -D cn=orcladmin -w
      password -v -f anon_off.ldif
      ```

7. Stop Oracle Internet Directory:

   ```
   (UNIX) ORACLE_HOME/bin/oidmon connect=db_connect_string stop
   (Windows) ORACLE_HOME\bin\oidmon connect=db_connect_string stop
   ```

8. Start OracleAS Infrastructure, including Oracle Internet Directory, in the Oracle Internet Directory Oracle home, then in any other OracleAS Infrastructure Oracle homes:

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

9. Start all middle tiers that are connected to the Infrastructure, as described in Section 3.2.1.

6.8.2 Enabling Anonymous Binds for Configuration Changes

If you have disabled anonymous binds, you must take the following steps to enable anonymous binds before you can make configuration changes to Oracle Application Server middle tiers or OracleAS Infrastructure:

1. Shut down all middle tiers that are connected to the OracleAS Infrastructure, as described in Section 3.2.2.

2. Shut down OracleAS Infrastructure, in all Infrastructure Oracle homes:

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

3. Start Oracle Internet Directory, because it must be started while you perform the procedure:

   ```
   (UNIX) ORACLE_HOME/bin/oidmon connect=db_connect_string start
   (Windows) ORACLE_HOME\bin\oidmon connect=db_connect_string start
   ```

4. Edit the `ias.properties` file for each middle tier connected to the OracleAS Infrastructure and for the Infrastructure Oracle home that contains OracleAS Single Sign-On and Oracle Delegated Administration Services. The `ias.properties` file is located in the following directory:

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

   In the `ias.properties` file, set the `OIDAnonymousDisabled` property to `false`:

   ```
   OIDAnonymousDisabled=false
   ```

   If the property does not exist in the file, or if it is set to `false`, anonymous binds are enabled.

5. Edit the `dads.conf` file for each middle tier connected to the OracleAS Infrastructure and for the Infrastructure Oracle home that contains OracleAS Single Sign-On and Oracle Delegated Administration Services. The `dads.conf` file is located in the following directory:

   ```
   (UNIX) ORACLE_HOME/Apache/modplsql/conf
   (Windows) ORACLE_HOME\Apache\modplsql\conf
   ```

   If you previously commented out the line that contains the PlsqlDatabaseConnectString parameter with a value that uses the LDAP name resolution format, uncomment out that line. If you deleted the line, add a line using the following format:

   ```
   PlsqlDatabaseConnectString cn=orcl, cn=oraclecontext NetServiceNameFormat
   ```

   If you previously added a line similar to the following, which contains the PlsqlDatabaseConnectString parameter with a value that use `host:port:service` format, comment out the line:

   ```
   PlsqlDatabaseConnectString db_host:db_hostdb_listener_port:db_service_name
   ```

6. Use the `ldapmodify` command to enable anonymous binds. Use the command on the Oracle home that contains Oracle Internet Directory.

   Take the following steps:

   a. Create a text file with the following lines:

```
dn:
changetype: modify
replace: orclanonymousbindsflag
orclanonymousbindsflag: 1
```

b. Use the `ldapmodify` command, calling the text file created in the previous step as input. In the following example, the text file is named `anon_on.ldif`:

```
(UNIX) ORACLE_HOME/bin/ldapmodify -h host -p port -D cn=orcladmin -w
password -v -f anon_on.ldif
(Windows) ORACLE_HOME\bin\ldapmodify -h host -p port -D cn=orcladmin -w
password -v -f anon_on.ldif
```

7. Stop Oracle Internet Directory:

```
(UNIX) ORACLE_HOME/bin/oidmon connect=db_connect_string stop
(Windows) ORACLE_HOME\bin\oidmon connect=db_connect_string stop
```

8. Start OracleAS Infrastructure, including Oracle Internet Directory, in the Oracle Internet Directory Oracle home, then in any other OracleAS Infrastructure Oracle homes:

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

9. Start all middle tiers that are connected to the Infrastructure, using the following command:

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

7

Changing Network Configurations

This chapter provides procedures for changing the network configuration of an Oracle Application Server host.

It contains the following topics:

- Overview of Procedures for Changing Network Configurations
- Changing the Hostname, Domain Name, or IP Address
- Moving Between Off-Network and On-Network
- Changing Between a Static IP Address and DHCP

7.1 Overview of Procedures for Changing Network Configurations

The following procedures for changing network configurations are presented in this chapter:

- Changing the Hostname, Domain Name, or IP Address

  This section describes how to update Oracle Application Server when changing the hostname, domain name, or IP address of a host.

- Moving Between Off-Network and On-Network

  This section provides procedures for moving an Oracle Application Server host on and off the network. You may use DHCP or a static IP address when on the network. You can use these procedures, for example, if you installed Oracle Application Server on your laptop and want to connect to different networks to use it.

- Changing Between a Static IP Address and DHCP

  This section provides procedures for changing from a static IP address to DHCP, and from DHCP to a static IP address. You might use these if you installed on a static IP address but then decide you want to use DHCP so you can be more mobile, or if you are using DHCP and must connect to a network using a static IP address.

If you have disabled anonymous binds in Oracle Internet Directory, you must enable them before you make configuration changes. See Section 6.8, "Disabling and Enabling Anonymous Binds" for more information.

7.2 Changing the Hostname, Domain Name, or IP Address

You may want to change the hostname, domain name, or IP address of the host after you have installed Oracle Application Server.

Table 7–1 summarizes the installation types that support hostname, domain name, and IP address changes, and provides pointers to the appropriate procedures.

Table 7–1   Supported Procedures for Hostname, Domain Name, and IP Address Changes

Installation Type	Changing the Hostname or Domain Name	Changing the IP Address
Middle tier	Supported See Section 7.2.2, "Changing the Hostname or Domain Name of a Middle-Tier Installation"	Supported Change the address in your operating system. No updates to Oracle Application Server are required.
10.1.4 or 10.1.2 Infrastructure: Identity Management only Identity Management installations with the following components configured: ■ Oracle Internet Directory only ■ OracleAS Single Sign-On, Oracle Delegated Administration Services, and (optionally) Oracle Directory Integration and Provisioning ■ Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, and (optionally) Oracle Directory Integration and Provisioning	Supported See Section 7.2.3, "Changing the Hostname, Domain Name, or IP Address of a 10.1.4 or 10.1.2 Identity Management Installation"	Supported See Section 7.2.3, "Changing the Hostname, Domain Name, or IP Address of a 10.1.4 or 10.1.2 Identity Management Installation"
10.1.4 or 10.1.2 Infrastructure: Identity Management and Metadata Repository	Not supported	Supported See Section 7.2.4, "Changing the IP Address of a 10.1.4 or 10.1.2 Infrastructure Containing a Metadata Repository"
10.1.4 or 10.1.2 Infrastructure: Metadata Repository only	Not supported	Supported See Section 7.2.4, "Changing the IP Address of a 10.1.4 or 10.1.2 Infrastructure Containing a Metadata Repository"

Many of the procedures in this section use the chgiphost command. See Section 7.2.1 for more information about the command.

7.2.1 Understanding the chgiphost Command

The chgiphost command-line utility changes the hostname, domain name, or IP address of a middle-tier instance, Infrastructure, or Identity Management installation.

The utility is located at:

■ On UNIX systems:

    ORACLE_HOME/chgip/scripts/chgiphost.sh

■ On Windows systems:

    ORACLE_HOME\chgip\scripts\chgiphost.bat

Table 7–2 shows the options for the command.

Table 7–2   Options for the chgiphost Command

Options	Description
-version	Displays the version of the utility
-mid	Changes the hostname, domain name, or IP address of a middle-tier instance
-silent	Runs the command in silent mode
-infra	Changes the IP address of a 10.1.4 or 10.1.2 Infrastructure instance. Does not apply to 10*g* Release 3 (10.1.3.4).
-idm	Changes the hostname, domain name, or IP address of a 10.1.4 or 10.1.2 Identity Management instance only. Does not apply to 10*g* Release 3 (10.1.3.4).

Note that if you use `chgiphost` to change the hostname or domain name, it does not update the instance name. For example, assume that the original instance name, with the hostname and domain name appended, is:

```
1013mid.myhost1.mydomain.com
```

If you change the hostname to myhost2, the instance name remains the same.

> **See Also:**
>
> - Section 7.2.5.1, "Setting the Log Level for chgiphost"
> - Section 7.2.5.2, "Customizing the chgiphost Command"

7.2.2 Changing the Hostname or Domain Name of a Middle-Tier Installation

This section describes how to change the hostname, domain name, or both, of a host that contains any of the middle-tier installation types.

The following sections describe the procedure:

- Before You Begin
- Task 1, "Export Oracle Enterprise Service Bus Services"
- Task 2, "Prepare Your Host"
- Task 3, "Change the Hostname"
- Task 4, "Run the chgiphost Command"
- Task 5, "Re-enable SSO Authentication"
- Task 6, "Update Host Information for Other Members of Cluster"
- Task 7, "Manually Update the Hostname in Files"
- Task 8, "Redeploy Oracle BPEL Process Manager Applications"
- Task 9, "Import Oracle Enterprise Service Bus Services"
- Task 10, "Restart Your Environment"

Before You Begin

Review the following items before you start:

- If the middle-tier instance is registered with Oracle Internet Directory, you must supply the `cn=orcladmin` password during the procedure.

- Consider changing the log level before running the `chgiphost` command so you can view more detailed information. See Section 7.2.5.1 for more information.

- If your old hostname is a string that is likely to appear in a configuration file, the `chgiphost` command may encounter problems when trying to update the configuration files. Refer to Section 7.2.5.2 for information on how to avoid this problem.

- Write down the old hostname and IP address before you begin. You will be prompted for these values.

- Oracle recommends that you perform a backup of your environment before you start this procedure. Refer to Part V, "Backup and Recovery" for more information.

- The procedure does not support Oracle Web Services Manager. The Oracle Web Services Manager files are not processed by the chgiphost script.

- The procedure does not support the Oracle BPEL Process Manager tutorials and demos, such as OrderBooking, if the tutorials and demos are set up to use a remote Oracle HTTP Server.

Task 1 Export Oracle Enterprise Service Bus Services

If Oracle Enterprise Service Bus is installed on your system, you must export the Oracle Enterprise Service Bus services before you change your network information. (You will import them later in this procedure.)

1. Run the following script, which sets the Oracle home and system variables:

   ```
   (UNIX) ORACLE_HOME/integration/esb/bin/esbdevprompt.sh
   (Windows) ORACLE_HOME\integration\esb\bin\esbdevprompt.bat
   ```

2. In the command-line window from which you ran the script, run the following command to export the Oracle Enterprise Service Bus services:

   ```
   ant export-params
   -Dparamfile ORACLE_HOME\integration\DateTimeStamp\esbparam.properties
   -DDB_URL=jdbc_connectString:@//hostname:port/db_service_name
   -DDB_USER=oraesb -DDB_PASSWORD=esb_pwd
   -DDB_DRIVER=oracle.jdbc.driver.OracleDriver
   ```

   In the example, *hostname* is the host containing the repository for Oracle Enterprise Service Bus, *port* is the port for the repository, *db_service_name* is the database service name, and *esb_pwd* is the password for the ORAESB user. For example, on Windows:

   ```
   ant export-params
   -Dparamfile ORACLE_HOME\integration\20060828_1503\esbparam.properties
   -DDB_URL=jdbc:oracle:thin:@//sta.oracle.com:1521/orcl.us.oracle.com
   -DDB_USER=oraesb -DDB_PASSWORD=ORAESB123
   -DDB_DRIVER=oracle.jdbc.driver.OracleDriver
   ```

Task 2 Prepare Your Host

Prepare your host for the change by stopping all processes:

1. Shut down each middle-tier instance on the host by running the following commands in each Oracle home:

   - On UNIX systems:

     ```
     ORACLE_HOME/opmn/bin/opmnctl stopall
     ```

   - On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl stopall
```

2. To make sure Oracle Application Server processes will not start automatically after a restart of the host, disable any automated startup scripts you may have set up, such as /etc/init.d scripts.

3. Make sure that the Oracle Internet Directory that is used by the middle tier is started.

Task 3 Change the Hostname

Update your operating system with the new hostname, domain name, or both. Consult your operating system documentation for information on how to perform the following steps. You can also change the IP address, if desired.

1. Make the updates to your operating system to properly change the hostname, domain name, or both.

2. Restart the host, if necessary for your operating system.

3. Verify that you can ping the host from another host in your network. Be sure to ping using the new hostname to make sure everything is resolving properly.

Task 4 Run the chgiphost Command

Follow these steps for each middle-tier instance on your host. Be sure to complete the steps entirely for one middle-tier instance before you move on to the next.

1. Log in to the host as the user that installed the middle-tier instance.

2. Make sure the ORACLE_HOME environment variable is set to the middle-tier Oracle home. Do not use a trailing slash (UNIX) or backslash (Windows) when specifying the variable.

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

4. Run the following commands in the middle-tier Oracle home:

   - On UNIX systems:

     ```
     cd ORACLE_HOME/chgip/scripts
     ./chgiphost.sh -mid
     ```

   - On Windows systems:

     ```
     cd ORACLE_HOME\chgip\scripts
     cmd /c chgiphost.bat -mid
     ```

   The chgiphost command prompts for information, as shown in Table 7–3. Note that the prompts may provide values in parentheses. You can enter a different value, or press the return key to accept the suggested value.

Table 7–3   Prompts and Actions for chgiphost -mid

Prompt	Action
`Enter fully qualified hostname (hostname.domainname) of destination`	Enter the new fully qualified hostname. This may be a new hostname, domain name, or both.
`Enter fully qualified hostname (hostname.domainname) of source`	Enter the original fully qualified hostname, domain name, or both.
`Enter valid IP Address of destination`	If you changed the IP address of the host, enter the new IP address. Otherwise, enter the current IP address.
`Enter valid IP Address of source`	If you changed the IP address of the host, enter the old IP address. Otherwise, enter the current IP address.
`OIDAdmin Password:`	Enter the `cn=orcladmin` password for the Oracle Internet Directory in which this middle-tier instance is registered.

5. Verify that the tool ran successfully by checking for errors in the files in the following directory:

   - On UNIX systems:

     ORACLE_HOME/chgip/log

   - On Windows systems:

     ORACLE_HOME\chgip\log

Task 5 Re-enable SSO Authentication

If the middle-tier instance was enabled for SSO authentication, you must enable it again. Follow the steps in "Task 1: Enable SSO Authentication (Optional)" in Section 6.7.

Task 6 Update Host Information for Other Members of Cluster

If the middle-tier instance is a member of a topology cluster, you may need to update the topology information in the opmn.xml file for the other members of the cluster.

If the middle-tier instance is part of a dynamic discovery cluster, you do not need to make any changes to the opmn.xml file.

However, if the cluster is one of the following types, you must update the opmn.xml file:

- Static hubs as discovery servers: You must change the host name or domain name in the <discover> element. The following example shows an example of the element:

```
<topology>
   <discover list="node1.com:6201,node2.com:6202"/>
</topology>
```

- Cross-topology gateways: You must change the host name or domain name in the <gateway> element. The following example shows an example of the element:

```
<topology>
 <gateway list="node1.com:6201&node2.com:6202&node3.com:6203"/>
 <discover list="*224.0.1.37:8205"/>
</topology>
```

- Static node-to-node: You must change the host name or domain name in the <nodes> element. The following example shows an example of the element:

```
<topology>
 <nodes list="node1-sun:6201,node2-sun:6202"/>
</topology>
```

Task 7 Manually Update the Hostname in Files

In some circumstances, you must manually update the hostname in files:

■   If you edited a file and entered the hostname as part of a user-defined parameter such as the Oracle home path, the hostname is not automatically updated by running the chgiphost command. To update the hostname in such cases, you must edit the files manually. For example on UNIX, the plsql.conf file may contain an NFS path including the hostname, such as: /net/dsun1/private/....

■   It is common that WebCenter applications use portlets that are running on the same host. For example, a WebCenter application running in OC4J instance OC4J_ Apps can consume a portlet running in the OC4J instance OC4J_WebCenter.

    If you change the hostname, domain name, or both, of a WebCenter application, you must manually update the internal references to these portlets. WebCenter applications store the portlet producer connection information in the file connections.xml, located in the following directory:

    ORACLE_HOME/j2ee/OC4J_instance/applications/application_name/adf/META-INF

    Using a text editor, search for the hostname and replace all occurrences with the appropriate new value. The information will look like that marked in bold in the following example:

```
<urlconnection name="PdkPortletProducer1_
1159964204445613411da8-010e-1000-8002-8c5707c5e057-urlconn"
url="http://apphost1.mycompany.com:8890/jpdk/providers"/>
.
.
.
<wsconnection
description="http://apphost1.mycompany.com:6688/richtextportlet/portlets/wsrp2?
WSDL">
    <soap
addressUrl="http://apphost1.mycompany.com:6688/richtextportlet/portlets/WSRP_
v2_PortletManagement_Service" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
```

    Similarly, if you are using Oracle Content DB and your connection information has changed, you can update connections.xml as follows:

```
<StringRefAddr addrType="jcr_oracle.ifs.jcr.configuration.serverUrl">
<Contents>https://ctdbhost1.mycompany.com:4444/content/ws</Contents></StringRef
Addr>
```

■   The chgiphost command also does not edit the hostname references in the documentation files. You must manually edit these files to update the hostname. Examples of such files are the index.html.* files in the *ORACLE_ HOME*/Apache/Apache/htdocs directory.

Task 8 Redeploy Oracle BPEL Process Manager Applications

If the instance contains Oracle BPEL Process Manager, you must redeploy the Oracle BPEL Process Manager applications:

1. Ensure that the Oracle BPEL Process Manager tmp directory, if it is exists in the cloned instance, is removed. The directory is located at:

```
(UNIX) ORACLE_HOME/bpel/domains/domainname/tmp/*
(Windows) ORACLE_HOME\bpel\domains\domainname\tmp\*
```

2. Restart the Oracle BPEL Process Manager server and redeploy all applications.

```
(UNIX) ORACLE_HOME/bpel/bin/startorabpel.sh
(Windows) ORACLE_HOME\bpel\bin\startorabpel.bat
```

Task 9 Import Oracle Enterprise Service Bus Services

If your instance contains Oracle Enterprise Service Bus, you must import the services from the file you exported in Task 1, "Export Oracle Enterprise Service Bus Services".

1. Edit the esbparam.properties file and update the properties to reflect the new environment. Change the DT_OC4J_HOST property to the new hostname, and if the HTTP port has changed, change the DT_OC4J_HTTP_PORT property. For example, if the new hostname is newhost.oracle.com, the value of the property would be:

```
DT_OC4J_HOST=newhost.oracle.com
```

2. Run the following command from the same command-line window you used in Task 1, "Export Oracle Enterprise Service Bus Services":

```
ant import-params
-Dparamfile ORACLE_HOME\integration\DateTimeStamp\esbparam.properties
-DDB_URL=jdbc_connectString:@//hostname:port/db_service_name
-DDB_USER=oraesb -DDB_PASSWORD=esb_pwd
-DDB_DRIVER=oracle.jdbc.driver.OracleDriver
```

In the example, *hostname* is the host containing the repository for Oracle Enterprise Service Bus, *port* is the port for the repository, *db_service_name* is the database service name, and *esb_pwd* is the password for the ORAESB user. For example, on Windows:

```
ant import-params
-Dparamfile ORACLE_HOME\integration\20060828_1503\esbparam.properties
-DDB_URL=jdbc:oracle:thin:@//sta.oracle.com:1521/orcl.us.oracle.com
-DDB_USER=oraesb -DDB_PASSWORD=ORAESB123
-DDB_DRIVER=oracle.jdbc.driver.OracleDriver
```

3. Redeploy all applications, including Oracle BPEL Process Manager applications.

Task 10 Restart Your Environment

Restart the middle-tier instances and restore your configuration to the way it was before you started the procedure:

1. Start each middle-tier instance on your host by running the following command in each Oracle home:

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

2. If you disabled any processes for automatically starting Oracle Application Server at the beginning of this procedure, enable them.

3. If your instance contains Oracle Enterprise Service Bus, verify that the services are listed in the ESB Console and that the applications run correctly.

7.2.3 Changing the Hostname, Domain Name, or IP Address of a 10.1.4 or 10.1.2 Identity Management Installation

You may have a 10*g* Release 3 (10.1.3.4) middle-tier instance associated with a 10.1.4 or 10.1.2 Identity Management installation.

This section describes how to change the hostname, domain name, or IP address on a host that contains an Identity Management installation. This procedure applies to any Identity Management-only installation, including the following:

- Identity Management with only Oracle Internet Directory configured

- Identity Management with OracleAS Single Sign-On, Oracle Delegated Administration Services, Oracle Directory Integration and Provisioning configured

- Identity Management with Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, Oracle Directory Integration and Provisioning configured

The following sections describe the procedure:

- Before You Begin

- Task 1: Shut Down Middle-Tier Instances

- Task 2: Prepare Your Host

- Task 3: Change the Hostname or IP Address

- Task 4: Run the chgiphost Command

- Task 5: Restart Your Environment

- Task 6: Update Your Environment

- Task 7: Update Oracle Internet Directory If LDAP-Based Replication Is Used

Before You Begin

Review the following items before you start the procedure:

- Consider changing the log level before running the `chgiphost` command so you can view more detailed information. See Section 7.2.5.1 for more information.

- If your old hostname is a string that is likely to appear in a configuration file, the `chgiphost` command may encounter problems when trying to update the configuration files. Refer to Section 7.2.5.2 for information on how to avoid this problem.

- Write down the old hostname and IP address before you begin. You will be prompted for these values.

- Oracle recommends that you perform a backup of your environment before you start this procedure. Refer to Part V, "Backup and Recovery" for more information.

Task 1: Shut Down Middle-Tier Instances

For each 10*g* Release 3 (10.1.3.4) middle-tier instance that uses Identity Management, stop Application Server Control and the middle-tier instance using the following command:

- On UNIX systems:

    ```
    ORACLE_HOME/opmn/bin/opmnctl stopall
    ```

- On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl stopall
```

Task 2: Prepare Your Host

Prepare your host for the hostname change by stopping all processes on the Release 2 (10.1.2) Identity Management instance:

1. Set the ORACLE_HOME environment variable.

2. Shut down the Identity Management installation, including the servers, such as Oracle Directory Server, Directory Integration and Provisioning Data server (if it is configured), and Replication Server, as well as Application Server Control. For example, on UNIX, use the following commands:

```
ORACLE_HOME/bin/emctl stop iasconsole
ORACLE_HOME/bin/oidctl server=odisrv instance=instance_number stop
ORACLE_HOME/bin/oidctl connect=global_db_name server=oidrepld
instance=instance_number stop
ORACLE_HOME/bin/oidctl server=oidldapd instance=instance_number stop
ORACLE_HOME/opmn/bin/opmnctl stopall
```

3. To make sure Oracle Application Server processes will not start automatically after a restart of the host, disable any automated startup scripts you may have set up, such as /etc/init.d scripts.

Task 3: Change the Hostname or IP Address

Update your operating system with the new hostname, domain name, or IP address. Consult your operating system documentation for information on how to perform the following steps:

1. Make the updates to your operating system to properly change hostname, domain name, or both.

2. Restart the host, if necessary for your operating system.

3. Verify that you can ping the host from another host in your network. Be sure to ping using the new hostname to make sure everything is resolving properly.

Task 4: Run the chgiphost Command

Perform these steps in the Identity Management instance:

1. Log in to the host as the user that installed Identity Management.

2. Set the ORACLE_HOME environment variable. Do not use a trailing slash (UNIX) or backslash (Windows) when specifying the ORACLE_HOME variable.

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

4. Run the following commands in the Identity Management Oracle home:

   - On UNIX systems:

   ```
   cd ORACLE_HOME/chgip/scripts
   ./chgiphost.sh -idm
   ```

   - On Windows systems:

   ```
   cd ORACLE_HOME\chgip\scripts
   ```

```
cmd /c chgiphost.bat -idm
```

The `chgiphost` command prompts for information, as shown in Table 7–4. Note that the prompts may provide values in parentheses. You can enter a different value, or press the return key to accept the suggested value.

Table 7–4 Prompts and Actions for chgiphost -idm

Prompt	Action
`Enter fully qualified hostname (hostname.domainname) of destination`	If you changed the hostname or domain name on your system, enter the new fully qualified hostname. Otherwise, enter the current fully qualified hostname.
`Enter fully qualified hostname (hostname.domainname) of source`	If you changed the hostname or domain name on your system, enter the old fully qualified hostname. Otherwise, enter the current fully qualified hostname.
`Enter valid IP Address of destination`	If you changed the IP address of the system, enter the new IP address. Otherwise, enter the current IP address
`Enter valid IP Address of source`	If you changed the IP address of the system, enter the old IP address. Otherwise, enter the current IP address

5. Verify that the tool ran successfully by checking for errors in the files in the following directory:

    ```
    (UNIX) ORACLE_HOME/chgip/log
    (Windows) ORACLE_HOME\chgip\log
    ```

Task 5: Restart Your Environment

Restart the Identity Management installation and any other Infrastructure instances that you stopped during this procedure:

1. Restart the Identity Management instance, using the following commands:

    ■   On UNIX systems:

    ```
    ORACLE_HOME/opmn/bin/opmnctl stopall
    ORACLE_HOME/opmn/bin/opmnctl startall
    ORACLE_HOME/bin/emctl start iasconsole
    ```

    ■   On Windows systems:

    ```
    ORACLE_HOME\opmn\bin\opmnctl stopall
    ORACLE_HOME\opmn\bin\opmnctl startall
    ORACLE_HOME\bin\emctl start iasconsole
    ```

2. If you disabled any processes for automatically starting Oracle Application Server at the beginning of this procedure, enable them.

Task 6: Update Your Environment

This task contains the steps to update your environment for the new hostname, domain name, or IP address. The steps you must take depend on how your environment is configured. If you changed the hostname or IP address of the host containing:

■   **Oracle Internet Directory only:** See "Configuration 1: Oracle Internet Directory Only". Oracle Internet Directory is installed on one host and the other Identity Management components are installed on another host and you change the host that contains Oracle Internet Directory. In this case, you must update the other

Identity Management components and the middle tiers that use this Identity Management.

- **Identity Management components other than Oracle Internet Directory:** See "Configuration 2: OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration and Provisioning". Oracle Internet Directory is installed on one host and the other Identity Management components are installed on another host and you change the host that contains the other Identity Management components. In this case, you must update the middle tiers that use this Identity Management.

- **Oracle Internet Directory and other Identity Management components:** See "Configuration 3: Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration and Provisioning". Oracle Internet Directory and the other Identity Management components are installed on the same host. In this case, you must update the middle tiers that use this Identity Management.

If your environment uses LDAP-based replication of Oracle Internet Directory and Oracle Internet Directory is on a different host than OracleAS Metadata Repository, you can change the hostname, domain name, or IP address of the host containing the Master (supplier) or Replica (consumer) Oracle Internet Directory. See "Task 7: Update Oracle Internet Directory If LDAP-Based Replication Is Used" on page 7-16 for information.

Configuration 1: Oracle Internet Directory Only In this case, Oracle Internet Directory is installed on one host and the other Identity Management components are installed on another host and you changed the host that contains Oracle Internet Directory. Take the following steps:

1. In the OracleAS Single Sign-On installation, stop the Infrastructure processes and Application Server Control:

   - On UNIX systems:

     ```
     ORACLE_HOME/opmn/bin/opmnctl stopall
     ORACLE_HOME/bin/emctl stop iasconsole
     ```

   - On Windows systems:

     ```
     ORACLE_HOME\opmn\bin\opmnctl stopall
     ORACLE_HOME\bin\emctl stop iasconsole
     ```

2. Update the `ias.properties` file in every OracleAS Infrastructure instance that uses Oracle Internet Directory. This includes other Identity Management instances (OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration and Provisioning, if it is configured).

   In each Oracle home, update the following file:

   ```
   (UNIX) ORACLE_HOME/config/ias.properties
   (Windows) ORACLE_HOME\config\ias.properties
   ```

   In the file, update the `OIDhost` parameter in with the new hostname:

   ```
   OIDhost=newhost.us.oracle.com
   ```

3. Update the `ldap.ora` file in every OracleAS Infrastructure instance that uses Oracle Internet Directory. This includes other Identity Management instances (OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration and Provisioning, if it is configured).

In each Oracle home, edit the following file:

```
(UNIX) ORACLE_HOME/ldap/admin/ldap.ora
(Windows) ORACLE_HOME\ldap\admin\ldap.ora
```

In the file, update the `DIRECTORY_SERVERS` parameter with the new fully qualified hostname.

4. In the Oracle homes for the other Identity Management components, start OPMN and Application Server Control:

- On UNIX systems:

  ```
  ORACLE_HOME/opmn/bin/opmnctl start
  ORACLE_HOME/bin/emctl start iasconsole
  ```

- On Windows systems:

  ```
  ORACLE_HOME\opmn\bin\opmnctl start
  ORACLE_HOME\bin\emctl start iasconsole
  ```

5. In the Oracle homes for the middle-tier instances, start OPMN and Application Server Control. To start Application Server Control, you start the default OC4J instance, because Application Server Control runs as an application with the default OC4J instance.

- On UNIX systems:

  ```
  ORACLE_HOME/opmn/bin/opmnctl start
  ORACLE_HOME/opmn/bin/opmnctl startproc process-type=home
  ```

- On Windows systems:

  ```
  ORACLE_HOME\opmn\bin\opmnctl start
  ORACLE_HOME\opmn\bin\opmnctl startproc process-type=home
  ```

6. If any middle-tier instance was enabled for SSO authentication, you must enable it again. Follow the steps in "Task 1: Enable SSO Authentication (Optional)" in Section 6.7 for each middle-tier instance that uses this Oracle Internet Directory instance.

7. In the Oracle homes for the other Identity Management components and each middle tier, run the Change Identity Management Services wizard and supply the new Oracle Internet Directory information. For example, for the 10.1.3 middle tier, take the following steps:

  a. Using Application Server Control, navigate to the OC4J Home page for the middle-tier instance.

  b. Click **Administration.**

  c. In the **Task Name** column of the table, expand **Security** if it is not already expanded. Then, in the Identity Management row, click the **Go to Task** icon.

  d. On the Identity Management page, click **Change.**

  e. Follow the steps in the wizard for supplying the new Identity Management information. See Section 6.7 for more information.

  f. When the operation completes, you must restart the OC4J instance. Do not click Restart on the Confirmation page. Instead, navigate to the Cluster Topology page, select the OC4J instance, and click **Restart.**

Note that although you may see the new Internet Directory host and port on the page, you still need to perform this step. For Release 2 (10.1.2), Application Server Control displays the virtual hostname only because it read it from the updated `ias.properties` file.

8. If OracleAS Certificate Authority is installed, take the following steps:

   a. Stop OracleAS Certificate Authority, the OC4J `oca` process, and the Oracle HTTP Server on the host running OracleAS Certificate Authority. For example, on UNIX, execute the following commands:

   ```
   ORACLE_HOME/oca/bin/ocactl stop
   ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=oca
   ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=HTTP_Server
   ```

   b. Edit the following file and change the name of the host listed in the file:

   ```
   (UNIX) ORACLE_HOME/oca/conf/oca.conf
   (Windows) ORACLE_HOME\oca\conf\oca.conf
   ```

   c. Reassociate with OracleAS Single Sign-On and Oracle Internet Directory. For example, on UNIX:

   ```
   ORACLE_HOME/oca/bin/ocactl changesecurity -server_auth_port OcaSslPort
   ```

   d. Start Oracle HTTP Server, the OC4J `oca` process, and OracleAS Certificate Authority. For example, on UNIX:

   ```
   ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=HTTP_Server
   ORACLE_HOME/opmn/bin/opmnctl startproc process-type=oca
   ORACLE_HOME/oca/bin/ocactl start
   ```

Configuration 2: OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration and Provisioning   In this case, Oracle Internet Directory is installed on one host and the other Identity Management components are installed on another host and you changed the host that contains the other Identity Management components.

In each middle-tier installation, take the following steps:

1. Start OPMN and Application Server Control. To start Application Server Control, you start the default OC4J instance, because Application Server Control runs as an application with the default OC4J instance.

   ■ On UNIX systems:

   ```
   ORACLE_HOME/opmn/bin/opmnctl start
   ORACLE_HOME/opmn/bin/opmnctl startproc process-type=home
   ```

   ■ On Windows systems:

   ```
   ORACLE_HOME\opmn\bin\opmnctl start
   ORACLE_HOME\opmn\bin\opmnctl startproc process-type=home
   ```

2. In the Oracle home for each middle tier, run the Change Identity Management Services wizard and supply the new Oracle Internet Directory information:

   a. Using Application Server Control, navigate to the OC4J Home page for the middle-tier instance.

   b. Click **Administration.**

   c. In the **Task Name** column of the table, expand **Security** if it is not already expanded. Then, in the Identity Management row, click the **Go to Task** icon.

d. On the Identity Management page, click **Change.**

e. Follow the steps in the wizard for supplying the new Identity Management information. See Section 6.7 for more information.

f. When the operation completes, you must restart the OC4J instance. Do not click Restart on the Confirmation page. Instead, navigate to the Cluster Topology page, select the OC4J instance, and click **Restart.**

Note that although you see the correct Internet Directory host and port on the page, you still need to perform this step.

3. Restart the affected components. Run the following commands in each Oracle home:

  – On UNIX systems:

```
ORACLE_HOME/opmn/bin/opmnctl stopall
ORACLE_HOME/opmn/bin/opmnctl startall
```

  – On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl stopall
ORACLE_HOME\opmn\bin\opmnctl startall
```

Configuration 3: Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration and Provisioning   In this case, Oracle Internet Directory and the other Identity Management components are installed on the same host and this is the host you changed. Take the following steps:

In each middle-tier installation, take the following steps:

1. Start OPMN and Application Server Control. To start Application Server Control, you start the default OC4J instance, because Application Server Control runs as an application with the default OC4J instance.

  ■ On UNIX systems:

```
ORACLE_HOME/opmn/bin/opmnctl start
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=home
```

  ■ On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl start
ORACLE_HOME\opmn\bin\opmnctl startproc process-type=home
```

2. If any middle-tier instance was enabled for SSO authentication, you must enable it again. Follow the steps in "Task 1: Enable SSO Authentication (Optional)" in Section 6.7 for each middle-tier instance.

3. In each middle-tier installation, run the Change Identity Management Services wizard:

a. Using Application Server Control, navigate to the OC4J Home page for the middle-tier instance.

b. Click **Administration.**

c. In the **Task Name** column of the table, expand **Security** if it is not already expanded. Then, in the Identity Management row, click the **Go to Task** icon.

d. On the Identity Management page, click **Change.**

    e. Follow the steps in the wizard for supplying the new Identity Management information. See Section 6.7 for more information.

    f. When the operation completes, you must restart the OC4J instance. Do not click Restart on the Confirmation page. Instead, navigate to the Cluster Topology page, select the OC4J instance, and click **Restart.**

4. Restart the affected components. Run the following commands in each Oracle home:

– On UNIX systems:

```
ORACLE_HOME/opmn/bin/opmnctl stopall
ORACLE_HOME/opmn/bin/opmnctl startall
```

– On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl stopall
ORACLE_HOME\opmn\bin\opmnctl startall
```

Task 7: Update Oracle Internet Directory If LDAP-Based Replication Is Used

If your environment uses LDAP-based replication of Oracle Internet Directory and Oracle Internet Directory is on a different host than OracleAS Metadata Repository, you can change the hostname, domain name, or IP address of the host containing the Master (supplier) or Replica (consumer) Oracle Internet Directory:

■ Configuration A: Host with Master Oracle Internet Directory Is Changed

■ Configuration B: Host with Replica Oracle Internet Directory Is Changed

Configuration A: Host with Master Oracle Internet Directory Is Changed

If you change the hostname, domain name, or IP address of the host containing the Master Oracle Internet Directory, take the following steps:

1. Obtain the replica ID of the Master Oracle Internet Directory:

```
ldapsearch -p master_port -h master_host -b "" -s base "objectclass=*"
orclreplicaid
```

2. On *both* the Master and the Replica, update either `orclreplicauri` or `orclreplicasecondaryuri` or both, if they exist, in the replica entry of the Master Oracle Internet Directory. Take the following steps:

    a. Create a file named `mod.ldif` and enter the following lines in the file:

```
dn: orclreplicaid=master_replicaID,cn=replication configuration
changetype:modify
replace: orclreplicauri
orclreplicauri: ldap://new_master_host:new_master_port/
```

    In the example, `master_replicaID` is the ID obtained in Step a, `new_master_host` is the new hostname of the Master Oracle Internet Directory, and `new_master_port` is the port number for the Master Oracle Internet Directory.

    b. Run the following command on the Master:

```
ldapmodify -p master_port -h master_host -f mod.ldif
```

    c. Run the following command on the Replica:

```
ldapmodify -p replica_port -h replica_host -f mod.ldif
```

3. Restart the Replication server at the Replica:

```
oidctl server=oidrepld inst=inst_num connect=connect_string flags="-h
 replica_host -p replica_port -m false" stop
oidctl server=oidrepld inst=inst_num connect=connect_string flags="-h
 replica_host -p replica_port -m false" start
```

In the example, *replica_host* is the hostname of the Replica Oracle Internet Directory and *replica_port* is the port of the Replica Oracle Internet Directory.

Configuration B: Host with Replica Oracle Internet Directory Is Changed

If you change the hostname, domain name, or IP address of the host containing the Replica Oracle Internet Directory, take the following steps:

1. Obtain the replica ID of the Replica Oracle Internet Directory:

```
ldapsearch -p replica_port -h replica_host -b "" -s base "objectclass=*"
orclreplicaid
```

2. On *both* the Master and the Replica, update either `orclreplicauri` or `orclreplicasecondaryuri` or both, if they exist, in the replica entry of the Replica Oracle Internet Directory. Take the following steps:

   a. Create a file named `mod.ldif` and enter the following lines in the file:

   ```
   dn: orclreplicaid=replica_replicaID,cn=replication configuration
   changetype:modify
   replace: orclreplicauri
   orclreplicauri: ldap://new_replica_host:new_replica_port/
   ```

   In the example, *replica_replicaID* is the ID obtained in Step a, *new_replica_host* is the new hostname of the Replica Oracle Internet Directory, and *new_replica_port* is the port number for the Replica Oracle Internet Directory.

   b. Run the following command on the Master:

   ```
   ldapmodify -p master_port -h master_host -f mod.ldif
   ```

   c. Run the following command on the Replica:

   ```
   ldapmodify -p replica_port -h replica_host -f mod.ldif
   ```

3. Restart the Replication server at the Replica:

```
oidctl server=oidrepld inst=inst_num connect=connect_string flags="-h
 new_replica_host -p new_replica_port -m false" stop
oidctl server=oidrepld inst=inst_num connect=connect_string flags="-h
 new_replica_host -p new_replica_port -m false" start
```

In the example, *new_replica_host* is the new hostname of the Replica Oracle Internet Directory and *new_replica_port* is the port of the Replica Oracle Internet Directory.

7.2.4 Changing the IP Address of a 10.1.4 or 10.1.2 Infrastructure Containing a Metadata Repository

This section describes how to change the IP address of a host that contains either of the following Infrastructure installation types:

- Metadata Repository only

- Identity Management and Metadata Repository

The following sections describe the procedure:

- Before You Begin
- Task 1: Shut Down Middle-Tier Instances
- Task 2: Prepare Your Host
- Task 3: Change the IP Address
- Task 4: Update the Infrastructure
- Task 5: Restart Your Environment

Before You Begin

Review the following items before you start the procedure:

- Write down the old IP address before you begin. You will be prompted for this during the procedure.

- Oracle recommends that you perform a backup of your environment before you start this procedure. Refer to Part V, "Backup and Recovery" for more information.

Task 1: Shut Down Middle-Tier Instances

Shut down all middle-tier instances that use the Infrastructure installation, even if they are on other hosts:

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

Task 2: Prepare Your Host

Prepare your host for the change by stopping all processes:

1. Set the ORACLE_HOME and ORACLE_SID environment variables.

2. Shut down the Infrastructure:

   a. Stop all Application Server Control and all processes:

      – On UNIX systems:

      ```
      ORACLE_HOME/bin/emctl stop iasconsole
      ORACLE_HOME/opmn/bin/opmnctl stopall
      ```

      – On Windows systems:

      ```
      ORACLE_HOME\bin\emctl stop iasconsole
      ORACLE_HOME\opmn\bin\opmnctl stopall
      ```

   b. Change directory to the `bin` subdirectory of Oracle home. Then, shut down the listener and database:

      ```
      lsnrctl stop

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

3. Verify that all Oracle Application Server processes have stopped.

4. To make sure Oracle Application Server processes will not start automatically after a restart of the host, disable any automated startup scripts you may have set up, such as /etc/init.d scripts.

Task 3: Change the IP Address

Update your operating system with the new IP address, restart the host, and verify that the host is functioning properly on your network. Consult your operating system documentation for information on how to perform the following steps:

1. Make the updates to your operating system to properly change the IP address.

2. Restart the host, if required by your operating system.

3. Verify that you can ping the host from another host in your network. Be sure to ping using the new IP address to make sure everything is resolving properly.

Task 4: Update the Infrastructure

Update the Infrastructure on your host with the new IP address:

1. Log in to the host as the user that installed the Infrastructure.

2. Set the ORACLE_HOME and ORACLE_SID environment variables. Do not use a trailing slash (UNIX) or backslash (Windows) when specifying the ORACLE_HOME variable.

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

4. Change directory to the bin subdirectory of Oracle home. Then, start the database and listener:

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

lsnrctl start
```

5. Start OPMN:

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

6. Start Oracle Internet Directory:

```
(UNIX) ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=OID
process-type=OID
(Windows) ORACLE_HOME\opmn\bin\opmnctl startproc ias-component=OID
process-type=OID
```

7. Run the following commands in the Infrastructure Oracle home:

   - On UNIX systems:

     ```
     cd ORACLE_HOME/chgip/scripts
     ./chgiphost.sh -infra
     ```

   - On Windows systems:

     ```
     cd ORACLE_HOME\chgip\scripts
     cmd /c chgiphost.bat -infra
     ```

The `chgiphost` command prompts for the old and new IP address.

8. Verify that the tool ran successfully by checking for errors in the files in the following directory:

```
(UNIX) ORACLE_HOME/chgip/log
(Windows) ORACLE_HOME\chgip\log
```

Task 5: Restart Your Environment

Start the remaining components of the Infrastructure and start any middle-tier instances that use it:

1. Start the Infrastructure:

   ■ On UNIX systems:

   ```
   ORACLE_HOME/opmn/bin/opmnctl startall
   ORACLE_HOME/bin/emctl start iasconsole
   ```

   ■ On Windows systems:

   ```
   ORACLE_HOME\opmn\bin\opmnctl startall
   ORACLE_HOME\bin\emctl start iasconsole
   ```

2. If a middle-tier instance is on the same host as the Infrastructure, then you must run the `chgiphost` command on the middle-tier instance before restarting the middle-tier instances.

3. Start the middle-tier instances:

   ■ On UNIX systems:

   ```
   ORACLE_HOME/opmn/bin/opmnctl startall
   ```

   ■ On Windows systems:

   ```
   ORACLE_HOME\opmn\bin\opmnctl startall
   ```

4. If you disabled any processes for automatically starting Oracle Application Server at the beginning of this procedure, enable them.

7.2.5 Special Topics for Changing a Hostname or Domain Name

This section contains the following special topics that apply to changing the hostname or domain name of an Oracle Application Server host:

■ Setting the Log Level for chgiphost

■ Customizing the chgiphost Command

■ Changing a Hostname After Upgrading from Windows 2000 to Windows 2003

■ Recovering from Errors When Changing a Hostname

7.2.5.1 Setting the Log Level for chgiphost

By default, the console log level for the `chgiphost` command is SEVERE. This causes only critical information to be printed while running `chgiphost`. To view additional progress information, set the console log level to CONFIG as follows:

1. Edit the following file:

   ```
   (UNIX) ORACLE_HOME/chgip/config/chgip.log.properties
   (Windows) ORACLE_HOME\chgip\config\chgip.log.properties
   ```

2. Change the `java.util.logging.ConsoleHandler.level` parameter to `CONFIG`:

```
java.util.logging.ConsoleHandler.level = CONFIG
```

7.2.5.2 Customizing the chgiphost Command

By default, the `chgiphost` command updates key configuration files in the Oracle home with the new hostname. If any of the following cases apply to your installation, you may want to consider customizing the behavior of the `chgiphost` command:

- You have created additional configuration files that contain the hostname and want the `chgiphost` command to update those files.

  To update these files, add their full path name to the following file before running `chgiphost`:

  ```
  (UNIX) ORACLE_HOME/chgip/config/hostname.lst
  (Windows) ORACLE_HOME\chgip\config\hostname.lst
  ```

- The old hostname is very short (one or two letters) or is a string that is likely to appear in a configuration file.

  Before running `chgiphost`, examine each of the files listed in `hostname.lst` to determine if the old hostname exists in any settings in those files. If you find a match, you can correct those settings after you run `chgiphost`.

- Your Oracle home contains the hostname in its full path.

  In this case, the `chgiphost` command may not update your configuration files properly. You can avoid this problem by using a Java utility called FileFixer, which searches for specific text strings in a file by matching regular expressions, and updates them to their new values. Note that FileFixer searches for patterns one line at a time. It cannot match patterns across lines.

  To use FileFixer:

  1. Make a copy of the following file:

  ```
  (UNIX) ORACLE_HOME/chgip/config/hostname_short_sample.lst.xml
  (Windows) ORACLE_HOME\chgip\config\hostname_short_sample.lst.xml
  ```

  2. Edit your copy of the file to specify the regular expression matching required for your old and new hostnames. The file contains an example of how to do this.

  3. Specify the file when running the `chgiphost` command:

  ```
  ./chgiphost option -hostnameShortXml full_path_to_your_xml_file
  ```

  For example, if you named your file `/mydir/my_sample.lst.xml`, and you are updating a middle-tier installation on UNIX, run `chgiphost` as follows:

  ```
  ./chgiphost -mid -hostnameShortXml /mydir/my_sample.lst.xml
  ```

7.2.5.3 Changing a Hostname After Upgrading from Windows 2000 to Windows 2003

When you upgrade from Windows 2000 to Windows 2003, lowercase letters in the hostname may be changed to uppercase letters. For example, if the hostname is `myhost` before the upgrade, it may be changed to `MYHOST`. If this occurs, some Oracle Application Server processes may not function properly.

To resolve this problem, you do not need to run the `chgiphost` command to update Oracle Application Server. You can simply add an entry with the lowercase hostname to the hosts file:

OS_path`\system32\drivers\etc\hosts`

For example, if the fully qualified hostname was `myhost.mydomain` before the upgrade, and the IP address is 1.2.3.4, add the following line:

`1.2.3.4 myhost.mydomain myhost`

7.2.5.4 Recovering from Errors When Changing a Hostname

This section describes how to recover from typical errors you might encounter when using the `chgiphost` command. It contains the following scenarios:

- Scenario 1: You Specified the Wrong Destination Name
- Scenario 2: You Encountered an Error When Running chgiphost

Scenario 1: You Specified the Wrong Destination Name

Suppose you ran the `chgiphost` command but specified the wrong destination name. In this case, you can remedy the error by running `chgiphost` again. Here are the details.

Suppose the current source hostname is loire985, the incorrect destination hostname you specified is mqa985, and the correct destination hostname is sqb985. Initially, you ran `chgiphost` with source = loire985 and destination = mqa985.

To recover from this error:

1. Run `chgiphost` with source = mqa985 and destination = sqb985.

2. Run `chgiphost` again with source = loire985 and destination = sqb985.

Scenario 2: You Encountered an Error When Running chgiphost

For example, you will get an error message if you enter the wrong password for Oracle Internet Directory. In this case, you should stop all processes in the instance using the `opmnctl stopall` command and run `chgiphost` again, with the same source and destination hostnames as before, and make sure to supply the correct password when prompted.

If you encounter an error when running `chgiphost`, you should fix the error, and run `chgiphost` again.

7.3 Moving Between Off-Network and On-Network

This section describes how to move an Oracle Application Server host on and off the network. The following assumptions and restrictions apply:

- The host must contain an instance that does not use an Infrastructure, or both the middle-tier instance and Infrastructure must be on the same host.

- DHCP must be used in loopback mode. Refer to *Oracle Application Server Installation Guide* for more information.

- Only IP address change is supported; the hostname must remain unchanged.

- Hosts in DHCP mode should not use the default hostname (`localhost.localdomain`). The hosts should be configured to use a standard hostname and the loopback IP should resolve to that hostname.

- A loopback adapter is required for all off-network installations (DHCP or static IP). Refer to *Oracle Application Server Installation Guide* for more information.

7.3.1 Moving from Off-Network to On-Network (Static IP Address)

This procedure assumes you have installed Oracle Application Server on a host that is off the network, using a standard hostname (not `localhost`), and would like to move on the network and use a static IP address. The IP address may be the default loopback IP, or any standard IP address.

To move onto the network, you can simply connect the host to the network. No updates to Oracle Application Server are required.

7.3.2 Moving from Off-Network to On-Network (DHCP)

This procedure assumes you have installed on a host that is off the network, using a standard hostname (not `localhost`), and would like to move on the network and use DHCP. The IP address of the host can be any static IP address or loopback IP address, and should be configured to the hostname.

To move onto the network:

1. Connect the host to the network using DHCP.

2. Configure the hostname to the loopback IP address only.

7.3.3 Moving from On-Network to Off-Network (Static IP Address)

Follow this procedure if your host is on the network, using a static IP address, and you would like to move it off the network:

1. Configure the `/etc/hosts` file so the IP address and hostname can be resolved locally.

2. Take the host off the network.

3. There is no need to perform any steps to change the hostname or IP address.

4. If the instance was configured as part of a cluster, you must remove it from the cluster before you start the instance. For example, to remove an instance from a cluster if you used dynamic discovery, use the following commands:

   - On UNIX:

     ```
     ORACLE_HOME/opmn/bin/opmnctl config topology delete discover
     ORACLE_HOME/opmn/bin/opmnctl reload
     ```

   - On Windows:

     ```
     ORACLE_HOME\opmn\bin\opmnctl config topology delete discover
     ORACLE_HOME\opmn\bin\opmnctl reload
     ```

   How you remove it from the cluster depends on how the cluster was configured. See the chapter, "Configuring and Managing Clusters" in the *Oracle Containers for J2EE Configuration and Administration Guide* for more information.

7.3.4 Moving from On-Network to Off-Network (DHCP)

Follow this procedure if your host is on the network, using DHCP in loopback mode, and you would like to move it off the network:

1. Take the host off the network.

2. There is no need to perform any steps to change the hostname or IP address.

3. If the instance was configured as part of a cluster, you must remove it from the cluster before you start the instance. For example, to remove an instance from a cluster if you used dynamic discovery, use the following commands:

   - On UNIX:

     ```
     ORACLE_HOME/opmn/bin/opmnctl config topology delete discover
     ORACLE_HOME/opmn/bin/opmnctl reload
     ```

   - On Windows:

     ```
     ORACLE_HOME\opmn\bin\opmnctl config topology delete discover
     ORACLE_HOME\opmn\bin\opmnctl config reload
     ```

   How you remove it from the cluster depends on how the cluster was configured. See the chapter, "Configuring and Managing Clusters" in the *Oracle Containers for J2EE Configuration and Administration Guide* for more information.

7.4 Changing Between a Static IP Address and DHCP

This section describes how to change between a static IP address and DHCP. The following assumptions and restrictions apply:

- The host must contain an instance that does not use an Infrastructure, or both the middle-tier instance and Infrastructure must be on the same host.

- DHCP must be used in loopback mode. Refer to *Oracle Application Server Installation Guide* for more information.

- Only IP address change is supported; the hostname must remain unchanged.

- Hosts in DHCP mode should not use the default hostname (`localhost.localdomain`). The hosts should be configured to use a standard hostname and the loopback IP should resolve to that hostname.

7.4.1 Changing from a Static IP Address to DHCP

To change a host from a static IP address to DHCP:

1. Configure the host to have a hostname 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 Application Server.

7.4.2 Changing from DHCP to a Static IP Address

To change a host from DHCP to a static IP address:

1. Configure the host to use a static IP address.

2. There is no need to update Oracle Application Server.

8

Changing Infrastructure Services

This chapter provides procedures for changing the Infrastructure Services used by a middle-tier instance.

It contains the following topics:

- Overview of Procedures for Changing Identity Management Services
- Changing Oracle Internet Directory from Dual Mode to SSL Mode
- Moving 10.1.4 or 10.1.2 Identity Management to a New Host

8.1 Overview of Procedures for Changing Identity Management Services

For this release, you can associate a 10*g* Release 3 (10.1.3.4) middle-tier instance with Release 10.1.4 or Release 2 (10.1.2) Identity Management Services, as described in Section 6.7.

After you have associated the middle-tier instance with Identity Management Services, you may want to change the Identity Management Services used by the middle-tier instance. For example, you may want to use an Identity Management Service on a different host.

You can change Identity Management Services using the Identity Management page of Application Server Control, shown in Figure 8–1.

Figure 8–1   Application Server Control Identity Management Page
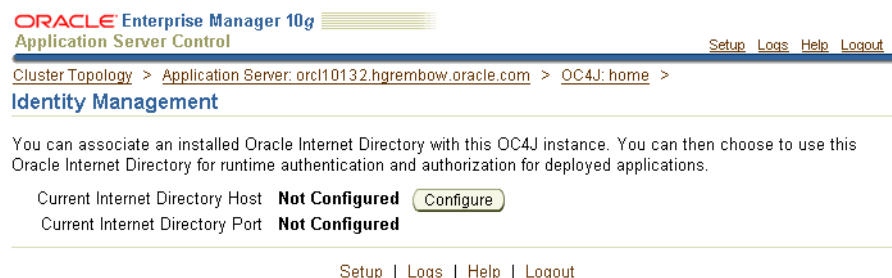
You must change Identity Management Services when you change any of the following:

- The HTTP OracleAS Single Sign-On port number on an Identity Management installation
- The Oracle Internet Directory non-SSL or SSL port number
- The Oracle Internet Directory Mode (Dual-mode or SSL)

- The host on which Identity Management is installed

If you have disabled anonymous binds in Oracle Internet Directory, you must enable them before you make configuration changes. See Section 6.8 for more information.

You cannot simply use the wizard to change from one Infrastructure service to another. You must first perform manual tasks to create and prepare the new Infrastructure service. This chapter describes the following supported procedures for changing Infrastructure Services:

- Changing Oracle Internet Directory from Dual Mode to SSL Mode

  Use this procedure if you want to change the Oracle Internet Directory mode from non-SSL to SSL. In addition to changing the mode, you must update middle-tier instances with the new mode, which requires changing Infrastructure Services.

- Moving 10.1.4 or 10.1.2 Identity Management to a New Host

  Use this procedure if you want to move your Identity Management installation, and its associated Metadata Repository, to a new host. After you perform the move, you must update middle-tier instances with the new host information for Identity Management, which requires changing Infrastructure Services.

For information about changing ports, see the following sections:

- To change the Oracle Internet Directory non-SSL or SSL port on an Identity Management installation, refer to Section 4.4.2 for instructions.

- To change the Oracle HTTP Server non-SSL or SSL Listen port on an Identity Management installation, which effectively changes the OracleAS Single Sign-On port, refer to Section 4.4.3 for instructions.

8.2 Changing Oracle Internet Directory from Dual Mode to SSL Mode

When you install Identity Management, you are asked to choose a mode for Oracle Internet Directory. The default mode is dual mode, which allows some components to access Oracle Internet Directory using non-SSL connections. During the installation, you can choose SSL mode, which specifies that all components must use SSL when connecting to the directory.

If you did not choose SSL mode during the installation, and want to change to SSL mode after installation, follow the procedure in this section. It includes changing the mode of the Oracle Internet Directory, and updating middle-tier instances to use the new mode.

8.2.1 Restrictions on Security Provider for Application Server Control

Before you begin this procedure, you must check that the security provider used by Application Server Control is a file-based security provider. If it is not, you must take additional steps after changing the Oracle Internet Directory mode.

To check the type of security provider, take the following steps:

1. In Application Server Control, navigate to the OC4J home page.

2. Click **Setup.**

3. On the Setup page, select **Security Provider.**

   The Security Provider page shows the type of security provider being used.

4. If it is not a file-based security provider and you want to change it, you can click **Change Security Provider.** Then, select **File-Based Security Provider** and specify a location for the XML file.

If the security provider is Oracle Internet Directory and you do not want to change it before running this procedure, you must take the steps in "Task 3: Change jazn.xml (Oracle Internet Directory Security Provider Only)".

8.2.2 Procedure

To change Oracle Internet Directory to SSL mode, perform the following tasks:

- Task 1: Stop Middle-Tier Processes and Start Application Server Control
- Task 2: Change the Oracle Internet Directory Mode
- Task 3: Change jazn.xml (Oracle Internet Directory Security Provider Only)
- Task 4: Change Middle-Tier Instances to Use SSL Mode

Task 1: Stop Middle-Tier Processes and Start Application Server Control

In all middle-tier instances that use Oracle Internet Directory, take the following steps:

1. Stop all middle-tier instances using the following command:

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

2. Because subsequent steps use Application Server Control, start OPMN and Application Server Control using the following commands. To start Application Server Control, you start the default OC4J instance, because Application Server Control runs as an application with the default OC4J instance.

   - On UNIX systems:

   ```
   ORACLE_HOME/opmn/bin/opmnctl start
   ORACLE_HOME/opmn/bin/opmnctl startproc process-type=home
   ```

   - On Windows systems:

   ```
   ORACLE_HOME\opmn\bin\opmnctl start
   ORACLE_HOME\opmn\bin\opmnctl startproc process-type=home
   ```

Task 2: Change the Oracle Internet Directory Mode

Perform this task on the Release 2 (10.1.2) Infrastructure that contains Oracle Internet Directory:

1. Create a file named mod.ldif and enter the following lines in the file:

```
dn:cn=configset0,cn=osdldapd,cn=subconfigsubentry
changetype:modify
replace:orclsslenable
orclsslenable:1
```

2. Run the following command:

```
ldapmodify -D "cn=orcladmin" -w orcladmin_passwd -p oid_port -v -f mod.ldif
```

In the example, `oid_port` is the non-SSL Oracle Internet Directory port. This is listed as OIDport in ORACLE_HOME/config/ias.properties.

Note that if you are using OracleAS Cold Failover Cluster, you must use the following command:

```
ldapmodify -D cn=orcladmin -w orcladmin_passwd -h virtual_hostname
-p oid_port -v -f mod.ldif
```

In the example, *virtual_hostname* is the virtual hostname of the OracleAS Cold Failover Cluster.

3. Stop the entire instance that contains Oracle Internet Directory:

   ■ On UNIX systems:

   ```
   ORACLE_HOME/bin/emctl stop iasconsole
   ORACLE_HOME/opmn/bin/opmnctl stopall
   ```

   ■ On Windows systems:

   ```
   ORACLE_HOME\bin\emctl stop iasconsole
   ORACLE_HOME\opmn\bin\opmnctl stopall
   ```

4. Edit the following file:

   ```
   (UNIX) ORACLE_HOME/ldap/admin/ldap.ora
   (Windows) ORACLE_HOME\ldap\admin\ldap.ora
   ```

   a. Modify the following line to remove the non-SSL port number:

   ```
   DIRECTORY_SERVERS=(myhost.myco.com:nonsslport:sslport)
   ```

   The resulting line looks similar to the following:

   ```
   DIRECTORY_SERVERS=(myhost.myco.com::sslport)
   ```

   b. Save and close the file.

5. If the OracleAS Metadata Repository was created using OracleAS RepCA, take the following steps:

   a. Copy the `ldap.ora` file from the Identity Management Oracle home to the Oracle home for the OracleAS Metadata Repository. For example, for Release 2 (10.1.2), the location is:

   ```
   (UNIX) ORACLE_HOME/ldap/admin
   (Windows) ORACLE_HOME\ldap\admin
   ```

   b. Edit the `sqlnet.ora` file that is located in the following location in the Oracle home for the OracleAS Metadata Repository:

   ```
   (UNIX) ORACLE_HOME/network/admin
   (Windows) ORACLE_HOME\network\admin
   ```

   Add LDAP to the NAMES.DIRECTORY_PATH entry, as shown in the following example:

   ```
   NAMES.DIRECTORY_PATH= (LDAP, TNSNAMES, ONAMES, HOSTNAME)
   ```

6. Edit the following file:

   ```
   (UNIX) ORACLE_HOME/config/ias.properties
   (Windows) ORACLE_HOME\config\ias.properties
   ```

   a. Change the `SSLOnly` parameter as follows:

   ```
   SSLOnly=true
   ```

   b. Save and close the file.

7. Restart the entire instance that contains Oracle Internet Directory:

- On UNIX systems:

```
ORACLE_HOME/opmn/bin/opmnctl stopall
ORACLE_HOME/bin/emctl stop iasconsole
ORACLE_HOME/opmn/bin/opmnctl startall
ORACLE_HOME/bin/emctl start iasconsole
```

- On Windows systems:

```
ORACLE_HOME\opmn\bin\opmnctl stopall
ORACLE_HOME\bin\emctl stop iasconsole
ORACLE_HOME\opmn\bin\opmnctl startall
ORACLE_HOME\bin\emctl start iasconsole
```

Task 3: Change jazn.xml (Oracle Internet Directory Security Provider Only)

If Oracle Internet Directory is the security provider for Application Server Control, you must make changes to `jazn.xml` for the instance that contains the active `ascontrol` application before you change that middle-tier instance to use SSL mode. (See Section 8.2.1 for information on determining the type of security provider used by Application Server Control.)

To make the changes, take the following steps for the instance that contains the active `ascontrol` application:

1. Edit the following file:

```
(UNIX) ORACLE_HOME/j2ee/OC4J_InstanceName/config/jazn.xml
(Windows) ORACLE_HOME\j2ee\OC4J_InstanceName\config\jazn.xml
```

2. Modify the `location` attribute to use the SSL port. For example:

```
location="ldap://myoid.us.oracle.com:636"
```

3. Modify the property value for `ldap.protocol` to be `ssl`. For example:

```
<property name=:ldap.protocol" value="ssl"/>
```

4. Save and close the file.

Task 4: Change Middle-Tier Instances to Use SSL Mode

In each middle-tier instance, run the Change Identity Management wizard and restart the instance:

1. Using Application Server Control, navigate to the OC4J Home page for the middle-tier instance.

2. Click **Administration.**

3. In the Task Name column of the table, expand **Security** if it is not already expanded. Then, in the Identity Management row, click the **Go to Task** icon.

4. On the Identity Management page, click **Change.**

5. On the Change Identity Management page:

- **Oracle Internet Directory Host:** Enter the fully qualified name of the Oracle Internet Directory host.

- **Oracle Internet Directory User DN:** Enter `cn=orcladmin`, or the distinguished name of a user in the `iASAdmins` group.

- **Password:** Enter the password for the user.

This password will be used as the default password for the oc4jadmin user created in Oracle Internet Directory.

- **Use only SSL connections to the Internet Directory:** Select this option.

    Then, in the **Oracle Internet Directory SSL Port** field, enter the Oracle Internet Directory SSL port number.

Click **OK.**

6. When the operation is finished, you must restart the OC4J instance. Do not click Restart on the Confirmation page. Instead, navigate to the Cluster Topology page, select the OC4J instance, and click **Restart.**

> **Note:** Now that you have disabled the non-SSL Oracle Internet Directory port, you must provide the "-U 1" option when using LDAP command-line utilities (such as ldapsearch, ldapmodify, and ldapaddmt) to connect to the SSL port.

8.3 Moving 10.1.4 or 10.1.2 Identity Management to a New Host

If you associated a 10*g* Release 3 (10.1.3.4) middle-tier instance with Release 10.1.4 or Release 2 (10.1.2) Identity Management Services, as described in Section 6.7, and you want to move Identity Management to a new host, follow the procedure in this section.

This procedure involves creating a replica (or copy) of the original Identity Management on a different host, along with its own new Metadata Repository, and then changing the middle-tier instance to use the new Identity Management.

See the *Oracle Application Server Upgrade and Compatibility Guide* for information about the specific versions of Oracle Identity Management that are supported with 10*g* Release 3 (10.1.3.4).

8.3.1 Sample Uses for This Procedure

The following are sample uses for this procedure:

- You have an existing Release 10.1.4 or Release 2 (10.1.2) Identity Management and associated Metadata Repository that is used by one or more 10*g* Release 3 (10.1.3.4) middle-tier instances. Your organization intends to replace the current Identity Management host with a new system. You can use this procedure to create a replica of the Identity Management, along with its own Metadata Repository, and change the middle-tier instances to use the new Identity Management. You can then retire the original host.

- You want to create a failover environment for your Release 10.1.4 or Release 2 (10.1.2) Identity Management. You can use this procedure to create a replica of the current Identity Management, along with its own Metadata Repository. You can keep the replica running so it stays synchronized with the original Identity Management. You can perform regular exports of data in the original Metadata Repository and save them. In the event that you lose the original Identity Management, you can import the data to the new Metadata Repository, and change the 10*g* Release 3 (10.1.3.4) middle-tier instances to use the new Identity Management. Refer to Section 8.3.4, "Strategy for Performing Failover with This Procedure" for more information.

8.3.2 Assumptions and Restrictions

- For both the original and new installations, the Identity Management and Metadata Repository can exist in the same Oracle home, or in separate Oracle homes (same or different host). If they are in separate Oracle homes, perform the operations on each in their own Oracle home.

- For both the original and new installations, the Identity Management components (OracleAS Single Sign-On, Oracle Internet Directory, Delegated Administration Services, Directory Integration and Provisioning) may exist in the same Oracle home, or may exist in separate Oracle homes (same or different host). If they exist in separate Oracle homes, perform the operations on each in their own Oracle home.

- This procedure does not take OracleAS Certificate Authority into consideration.

  > **See Also:** *Oracle Application Server Certificate Authority Administrator's Guide* for information on updating OracleAS Certificate Authority when changing Identity Management services

8.3.3 Procedure for Moving Identity Management to a New Host

This section describes how to move a Release 10.1.4 or Release 2 (10.1.2) Identity Management to a new host.

The following presents an overview of the procedure:

1. You have an original Release 10.1.4 or Release 2 (10.1.2) Identity Management (also called the Master) used by one or more middle-tier instances. The Identity Management has a Metadata Repository. You install and set up a new Identity Management (also called the Replica). This Identity Management has its own Metadata Repository. The Oracle Internet Directory in the new Identity Management is an LDAP-based replica of the original Oracle Internet Directory. Replication takes place constantly from the original Oracle Internet Directory to the new Oracle Internet Directory.

Figure 8–2 shows this setup with a Release 2 (10.1.2) Identity Management.

Figure 8–2   Original Host (Master) and New Host (Replica)
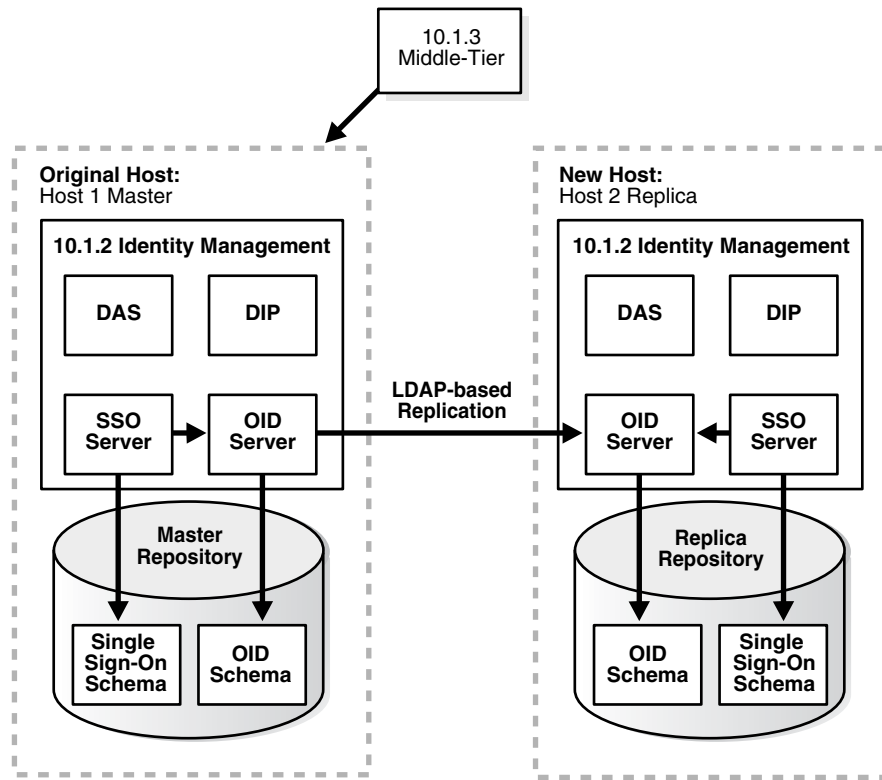
See: "Task 1: Install and Set Up the New Identity Management and Metadata Repository"

2. You migrate OracleAS Single Sign-On and Directory Integration and Provisioning data from the original Metadata Repository (Master) to the new Metadata Repository (Replica).

   See: "Task 2: Migrate OracleAS Single Sign-On and Directory Integration and Provisioning Data"

3. You change the middle-tier instances to use the new Identity Management.

   See: "Task 3: Change Middle-Tier Instances to the New Identity Management"

4. You stop the LDAP-based replication.

   See: "Task 4: Stop Replication"

Figure 8–3 illustrates the steps described.

Figure 8–3   Changing from Original to New Identity Management
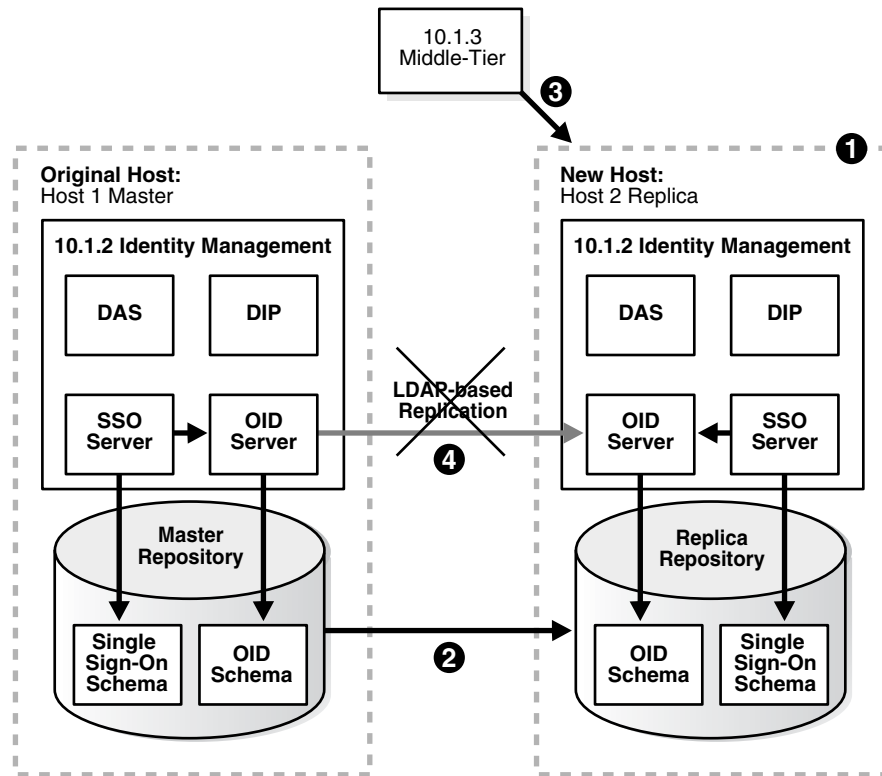
Task 1: Install and Set Up the New Identity Management and Metadata Repository

In this task, you install and set up the new Release 10.1.4 or Release 2 (10.1.2) Identity Management and its associated Metadata Repository. The new Identity Management is an LDAP-based replica of the original Identity Management.

1. Read Section F.1, "About LDAP-Based Replicas" to learn about LDAP-based replicas and how they are used for this procedure.

2. Follow the procedure in Section F.2, "Installing and Setting Up an LDAP-Based Replica" to install and set up the new Identity Management and Metadata Repository.

Task 2: Migrate OracleAS Single Sign-On and Directory Integration and Provisioning Data

In this task, you migrate the OracleAS Single Sign-On and Directory Integration and Provisioning Data from the original Metadata Repository to the new Metadata Repository. The source for the migration is the original Metadata Repository (Master) and the target for the migration is the new Metadata Repository (Replica).

This task contains the following subtasks:

■   Migrate the OracleAS Single Sign-On Data

■   Migrate the Directory Integration and Provisioning Data

> **Note:**   Make sure the ORACLE_HOME and ORACLE_SID environment variables are set before you begin. This applies to all platforms.

Migrate the OracleAS Single Sign-On Data

To migrate the OracleAS Single Sign-On data:

1. Obtain the ORASSO schema password on the master:

   ```
   MASTER_HOME/bin/ldapsearch -p master_oid_port -h master_host
    -D "cn=orcladmin" -w master_orcladmin_passwd
    -b "orclresourcename=orasso,  orclreferencename=master_global_db_name,
   cn=ias infrastructure databases, cn=ias, cn=products, cn=oraclecontext"
   -s base "objectclass=*" orclpasswordattribute
   ```

   This command prints the ORASSO password in a line similar to the following:

   ```
   orclpasswordattribute=LAetjdQ5
   ```

2. Export the OracleAS Single Sign-On data from the master, ensuring that the ORACLE_HOME environment variable is set before you run this command:

   ```
   MASTER_HOME/sso/bin/ssomig -export -s orasso -p master_orasso_passwd
   -c master_db_name -log_d $MASTER_HOME/sso/log
   ```

   In the example, `master_orasso_passwd` is the ORASSO password obtained in the previous step.

3. Copy the `ssomig.dmp` and `ssoconf.log` files from the master to the replica, preserving the exact full path for each file.

4. Obtain the ORASSO schema password on the replica:

   ```
   REPLICA_HOME/bin/ldapsearch -p replica_oid_port -h replica_host
   -D "cn=orcladmin" -w replica_orcladmin_password -b "orclresourcename=orasso,
   orclreferencename=replica_global_db_name, cn=ias infrastructure databases,
   cn=ias, cn=products, cn=oraclecontext" -s base "objectclass=*"
   orclpasswordattribute
   ```

5. Import the OracleAS Single Sign-On data to the replica:

   ```
   REPLICA_HOME/sso/bin/ssomig -import -overwrite -s orasso
   -p replica_orasso_passwd -c replica_db_name
   -log_d $REPLICA_HOME/sso/log -discoforce
   ```

   In the example, `replica_orasso_passwd` is the ORASSO password obtained in the previous step.

6. Verify that the export and import of OracleAS Single Sign-On succeeded.

   Verify that the OracleAS Single Sign-On migration tool reported success. You can also check the following log files for errors:

   ```
   MASTER_HOME/sso/log/ssomig.log
   REPLICA_HOME/sso/log/ssomig.log
   ```

   > **See Also:** *Oracle Application Server Single Sign-On Administrator's Guide*, Release 2 (10.1.2) for information on interpreting messages in the log files

7. Reenable SSO authentication, as described in "Task 1: Enable SSO Authentication (Optional)" in Section 6.7.

Migrate the Directory Integration and Provisioning Data

To migrate your Directory Integration and Provisioning Data:

> **See Also:** Directory Integration and Provisioning Data
> documentation in the *Oracle Internet Directory Administrator's Guide*,
> Release 2 (10.1.2) for information about running the following
> commands using the HTTPS port in environments in which the Oracle
> Internet Directory HTTP port is disabled

1. Stop the Directory Integration and Provisioning Data server on the master:

   ```
   MASTER_HOME/bin/oidctl server=odisrv instance=1 stop
   ```

2. Migrate the Directory Integration and Provisioning Data:

   ```
   MASTER_HOME/bin/dipassistant reassociate -src_ldap_host master_host
   -src_ldap_port master_oid_port -dst_ldap_host replica_host
   -dst_ldap_port replica_oid_port -src_ldap_passwd master_orcladmin_passwd
   -dst_ldap_passwd replica_orcladmin_passwd
   ```

   This command prints log messages to:

   ```
   MASTER_HOME/ldap/odi/log/reassociate.log
   ```

3. Stop the Directory Integration and Provisioning Data server on the replica:

   ```
   REPLICA_HOME/bin/oidctl server=odisrv instance=1 stop
   ```

4. Register the Directory Integration and Provisioning Data server on the replica:

   ```
   REPLICA_HOME/bin/odisrvreg -D "cn=orcladmin" -w replica_orcladmin_passwd
   -h replica_host -p replica_oid_port
   ```

5. Start the Directory Integration and Provisioning Data server on the replica:

   ```
   REPLICA_HOME/bin/oidctl server=odisrv instance=1 flags="port=replica_oid_port"
   start
   ```

Task 3: Change Middle-Tier Instances to the New Identity Management

In each middle-tier instance, take the following steps:

1. Using Application Server Control, navigate to the OC4J Home page for the
   middle-tier instance.

2. Click **Administration.**

3. In the **Task Name** column of the table, expand **Security** if it is not already
   expanded. Then, in the Identity Management row, click the **Go to Task** icon.

4. On the Identity Management page, click **Change.**

5. Follow the steps in the wizard for supplying the new Identity Management
   information. See Section 6.7 for more information.

6. When the operation is finished, you must restart the OC4J instance. Do not click
   Restart on the Confirmation page. Instead, navigate to the Cluster Topology page,
   select the OC4J instance, and click **Restart.**

If you have a problem changing the middle-tier instances to the new host, check to
make sure replication is running and try again.

Task 4: Stop Replication

Stop the replication between the original Identity Management and the new Identity
Management (Replica) by running the following command in the new Identity
Management Oracle home:

```
oidctl connect=global_db_name server=oidrepld instance=1 flags="-p oid_port" stop
```

In the example:

- *global_db_name* is the global database name of the new Identity Management. (This is referred to as *replica_db_name* in Section F.2, "Installing and Setting Up an LDAP-Based Replica".)

- *oid_port* is the non-SSL Oracle Internet Directory port in the new Identity Management.

8.3.4 Strategy for Performing Failover with This Procedure

As mentioned in Section 8.3.1, you can modify this procedure to perform failover for Identity Management. This enables you to move your middle-tier instances to the new Identity Management in case the original is lost.

Figure 8–4   Failover to New Identity Management
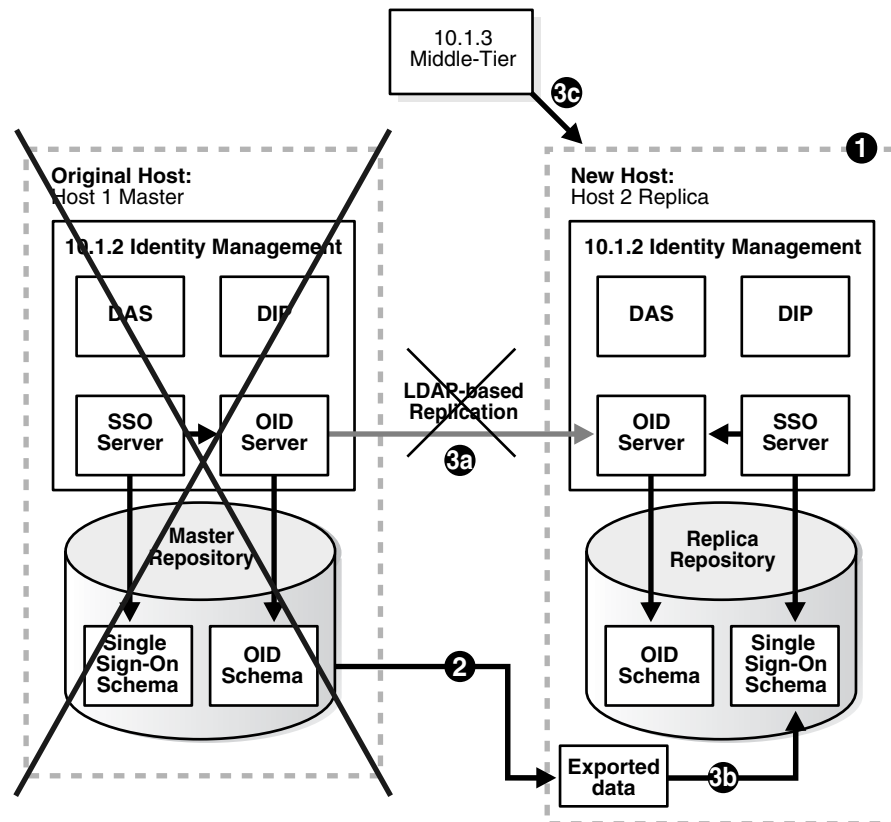
To set up the environment for failover:

1. Install and set up the new Identity Management as described in "Task 1: Install and Set Up the New Identity Management and Metadata Repository".

2. Export OracleAS Single Sign-On and Directory Integration and Provisioning data on a regular basis from the original Metadata Repository. You do not need to import the data into the new Metadata Repository. You only need to export the data and copy the files to the new Metadata Repository host. Refer to "Task 2: Migrate OracleAS Single Sign-On and Directory Integration and Provisioning Data".

3. If you lose the original Identity Management:

a. Stop replication, as described in "Task 4: Stop Replication".

b. Import your most recent copy of the OracleAS Single Sign-On and Directory Integration and Provisioning data into the new Identity Management repository. Refer to "Task 2: Migrate OracleAS Single Sign-On and Directory Integration and Provisioning Data".

c. Change the middle-tier instances to use the new Identity Management as described in "Task 3: Change Middle-Tier Instances to the New Identity Management".

9

Cloning Application Server Middle-Tier Instances

This chapter provides information on cloning an installation of an Oracle Application Server middle-tier instance.

It contains the following topics:

- Introduction to Cloning
- What Installation Types Can You Clone?
- Understanding the Cloning Process
- Cloning Oracle Application Server Instances
- Considerations and Limitations for Cloning
- Customizing the Cloning Process
- Example: Using Cloning to Expand an Oracle Application Server Cluster

9.1 Introduction to Cloning

Cloning is the process of copying an existing installation to a different location while preserving its configuration. Some situations in which cloning an installation of Oracle Application Server is useful are:

- Creating an installation that is a copy of a production, test, or development installation. Cloning enables you create a new installation with all patches applied to it in a single step. This is in contrast to separately installing, configuring and applying any patches to Oracle Application Server.

- Rapidly deploying an instance and the applications it hosts.

- Preparing a "gold" image of a patched home and deploying it to many hosts.

The cloned installation behaves the same as the source installation. For example, the cloned instance can be deinstalled or patched using Oracle Universal Installer. It can also be used as the source for another cloning operation.

You can create a cloned copy of a test, development, or production installation by using the command-line cloning scripts.

The default cloning procedure is adequate for most usage cases. However, you can also customize various aspects of the cloning process, for example, to specify custom port assignments, or preserve custom settings.

Figure 9–1 shows cloning a J2EE Server and Web Server middle tier that is not connected to Oracle Identity Management.

Figure 9–1   Cloning a J2EE Server and Web Server Middle Tier
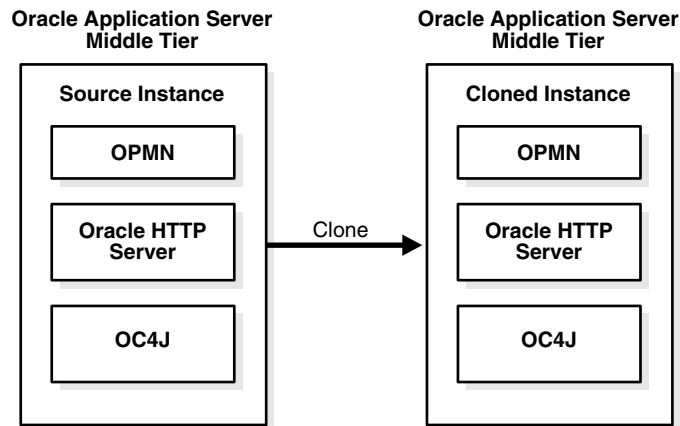
The cloning process works by copying all files from the source Oracle home to the destination Oracle home. Hence, any files used by the source instance that are located outside the source Oracle home's directory structure are not copied to the destination location.

After the files are copied, a set of scripts is used to update the information in key configuration files. For example, all references to the host name and the Oracle home in `httpd.conf` are updated to their new values.

Any applications deployed in the source instance are also copied to the cloned instance and automatically deployed, provided they are located in the source Oracle home's directory structure.

9.2 What Installation Types Can You Clone?

In this release, you can clone the following types of middle-tier installations:

- Basic Installation: J2EE Server and SOA Suite

- J2EE Server, Web Server, and SOA Suite

- J2EE Server and Web Server

- J2EE Server

- Web Server (Oracle HTTP Server)

- Oracle WebCenter Framework

- Oracle WebCenter Framework with Oracle HTTP Server

See Section 9.5 for details of considerations and limitations affecting specific components in the cloned Oracle home.

Note the following:

- You cannot clone an instance that contains Oracle Content DB. However, you can clone an instance that uses Oracle Content DB (in a different instance) as a repository for Oracle WebCenter Framework.

- You cannot clone Oracle TopLink.

- You cannot clone Oracle Web Services Manager. The cloning operation copies the files from the source to the cloned instance, but you cannot use the cloned Oracle Web Services Manager instance.

- You cannot clone an OracleAS Infrastructure, but you can clone a middle tier that is connected to Oracle Identity Management. However, the cloned instance is not associated with Oracle Identity Management; you must manually associate it with Oracle Identity Management. See Section 6.7, "Configuring Instances to Use 10.1.4 or 10.1.2 Oracle Identity Management" for instructions.

- You cannot clone an instance that is connected to Oracle Database Lite, such as the Basic Install on Windows.

- The cloned instance must have a different instance name than the source instance. You specify the instance name when you clone the instance, as described in Section 9.4.3.

- You can clone a middle-tier instance that is a member of a cluster, but only if it is based on multicast discovery or static node discovery. See Section 9.4.5 for more information.

9.3 Understanding the Cloning Process

The cloning process makes use of the cloning functionality in Oracle Universal Installer. The operation is driven by a set of scripts that are included in the Oracle Application Server installation. The following sections describe the processes involved in cloning an instance:

1. Source Preparation Phase
2. Cloning Phases

9.3.1 Source Preparation Phase

At the source, you run the script called `prepare_clone.pl`. This is a Perl script that prepares the source for cloning. It takes a snapshot of the information required for cloning.

During this phase, `prepare_clone.pl` parses files in the source Oracle home to extract and store required values and backs up required files.

Then, you tar the Oracle home directories.

See Section 9.4.2 for specific instructions for preparing the source instance.

9.3.2 Cloning Phases

At the destination, you extract the Oracle home from the tar file. Then, you run the script called `clone.pl`. This is a Perl script that performs all parts of the cloning operation automatically, calling various other utilities and Oracle Universal Installer, as needed. When you invoke the `clone.pl` script, it goes through the following three phases:

1. Pre-cloning phase

   During this phase, the `clone.pl` script lays the groundwork necessary to ensure that cloning can be done.

2. Cloning phase

   During this phase, the `clone.pl` script invokes Oracle Universal Installer in clone mode with the necessary arguments to perform the Oracle Universal Installer home cloning. This re-instantiates all files (after creating backups of the existing instantiated files), sets environment variables, updates links, and so on. In other

words, it repeats all actions that were performed at installation time, with the exception of the file copying.

3. Post-cloning phase

The postinstallation configuration assistants are not designed to be run again at clone time. Consequently, some of the instance-specific configuration files that should be updated by the configuration assistants are not updated at the end of the Oracle Universal Installer cloning session. Instead, Oracle has created a set of post-cloning scripts that update those files to bring the cloned home to a working state.

The post-cloning steps performed by the script are:

a. Sets the new Oracle home.

b. Updates configuration files. In this step, many configuration files that have been re-instantiated by Oracle Universal Installer during the cloning phase are restored from their backups. Those files are then updated with the new values that reflect the new environment, if needed. For example, if a file has a reference to the source Oracle home, that reference is updated to reflect the destination Oracle home.

c. Calls the home's `chgiphost` command to change the host name and IP number information in the cloned home. Before calling `chgiphost`, the script must collect the following required information to invoke `chgiphost` in silent mode:

   – Source host name

   – Source IP address

   – Destination host name

   – Destination IP address

   Note that when `chgiphost` is run as part of cloning, it does not run all of the configuration tools that are run when `chgiphost` is run standalone (as when you change hostname or domain name).

d. If the source instance is connected to Oracle Internet Directory, adds information about the clone to Oracle Internet Directory.

e. After all the operations for cloning are completed, starts services, as well as Application Server Control, to verify the success of the cloning operation.

Note that you do not need to perform each of these phases manually, because the `clone.pl` script takes care of all three phases automatically. The information is provided only for conceptual understanding.

See Section 9.4.3 for specific instructions on the tasks you do at the destination.

Files Updated During the Post-Cloning Phase

During the post-cloning phase, a set of important configuration files are restored from their backup versions and updated. Typical changes made to the files include updating environment-specific variables such as host name, Oracle home, and port numbers to their new values.

The following list shows some of the key files that are updated. Note that this is not an exhaustive list of the files being updated.

- *Oracle_Home*/config/ias.properties

- *Oracle_ Home*/sysman/j2ee/application-deployments/ascontrol/orion-web. xml

- *Oracle_Home*/Apache/Apache/conf/httpd.conf

- *Oracle_Home*/Apache/Apache/conf/mod_oc4j.conf

- *Oracle_Home*/Apache/Apache/conf/oracle_apache.conf

- *Oracle_Home*/Apache/modplsql/conf/dads.conf

- *Oracle_Home*/Apache/modplsql/conf/plsql.conf

- *Oracle_Home*/Apache/modplsql/conf/cache.conf

- *Oracle_Home*/Apache/oradav/conf/moddav.conf

- *Oracle_Home*/opmn/conf/opmn.xml

- *Oracle_Home*/backup_restore/config/config_misc_files.inp

The format of the paths are shown in UNIX format. For Windows, invert the slashes.

9.4 Cloning Oracle Application Server Instances

To clone an Oracle Application Server instance, you copy the scripts from the Companion CD. Then, you first prepare the source Oracle home and then you clone the destination.

9.4.1 Prerequisites for Cloning

For cloning, Perl 5.83 or higher must be installed on your system. Before running the cloning Perl scripts, you must set the PERL5LIB environment variable to the path of the Perl directory in the Oracle home. This path must be the first one listed in the variable definition. For example:

- On UNIX

```
export PERL5LIB=$ORACLE_HOME/perl/lib/5.8.3/i686-linux-thread-multi:$ORACLE_
HOME/perl/lib/5.8.3:$ORACLE_HOME/perl/lib/site_
perl/5.8.3/i686-linux-thread-multi/
```

- On Windows:

```
set PERl5LIB=%ORACLE_HOME%\perl\5.8.3\lib;%ORACLE_
HOME%\perl\5.8.3\lib\MSWin32-x86-multi-thread;%ORACLE_
HOME%\perl\site\5.6.1\lib;%ORACLE_HOME%\perl\site\5.8.3\lib
```

9.4.2 Preparing the Source

To prepare the source Oracle home to be cloned, take the following steps at the source instance:

1. If the source instance contains Oracle Enterprise Service Bus, export the services as described in Section 9.5.6.

2. Change to the following directory:

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

3. Run the script prepare_clone.pl. This script prepares the source to be cloned.

The command line for the script has the following format:

```
perl prepare_clone.pl [ORACLE_HOME=OH_dir]
                      [-silent]
                      [-debug]
                      [-export]
                      [-help]
```

In the example, substitute the following for *perl*:

- On UNIX:

    ```
    $ORACLE_HOME/perl/bin/perl
    ```

- On Windows:

    ```
    %ORACLE_HOME%\perl\5.8.3\bin\MSWin32-x86-multi-thread\perl5.8.3
    ```

Table 9–1 describes the parameters and options for the `prepare_clone.pl` script.

Table 9–1   Parameters and Options for the prepare_clone.pl Script

Parameter or Option	Description
ORACLE_HOME	The complete directory specification for the source Oracle home. If you do not supply this parameter, the script uses the ORACLE_HOME environment variable, if it exists. If the environment variable does not exist, the script assumes that ORACLE_HOME is the directory from which the script is being run.
	Do not use a trailing slash (UNIX) or backslash (Windows) when specifying the Oracle home.
	Use the value that was provided during installation; do not use symbolic links.
	If ORACLE_HOME is invalid, the script exits and logs an error to standard output (STDOUT).
-silent	Runs the script in silent mode. If the command line does not contain the required password-related options, the script exits.
-debug	Runs the script in debug mode.
-export	Exports the page customizations and portlet metadata that are stored in MDS on the source instance to an .ear file. Also exports the portlet customizations (preferences) to the .ear file. Use this option to migrate customizations that are associated to a WebCenter application from one location to another if the cloned instance will use a different MDS location.
	This option calls the export mode of the Oracle WebCenter Framework Predeployment tool. For more information about the Predeployment tool, see "Deploying Your WebCenter Application" in the *Oracle WebCenter Framework Developer's Guide*.
	The script creates an .ear file with the suffix `_clone_export.ear` in the following directory:
	(UNIX) *ORACLE_HOME*/j2ee/*instance*/applications/*app_name*/*app_name*_clone_export.ear (Windows) *ORACLE_HOME*\j2ee*instance*\applications*app_name**app_name*_clone_export.ear
-help	Prints the usage for the script.

4. Archive and compress the source Oracle home, using your preferred tool for archiving. For example, you can use WinZip on Windows and tar and gzip on

UNIX. Make sure that the tool you are using preserves the permissions and timestamps of the files. The following example shows how to archive and compress the source on UNIX:

```
cd Source_Oracle_Home
tar cf - * | gzip > oracleas.tar.gz
```

The tar utility may issue warnings if the sticky bit is set on some files. You can safely ignore these warnings.

Note that you should not use the `jar` utility to archive and compress the Oracle home.

9.4.3 Cloning the Instance

At the destination, to clone the source instance, take the following steps:

1. Copy the compressed Oracle home from the source machine to the destination machine.

2. Extract the compressed Oracle home into a directory, which will become the new Oracle home at the destination location. Use your preferred tool to extract the compressed files. For example, you can use WinZip on Windows and tar and gunzip on UNIX. Make sure that the tool you are using preserves the permissions and timestamps of the files. The following example shows how to extract the files on UNIX:

    ```
    mkdir -p Destination_Oracle_Home
    cd Destination_Oracle_Home
    gunzip < Dir_Containing_Tar/oracleas.tar.gz | tar xf -
    ```

    > **Note:** Make sure that the tar and gzip/gunzip versions on the source and destination machines are compatible. You may encounter problems unzipping the archive if these versions differ.

3. Change to the following directory:

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

4. Run the `clone.pl` script. You must have write permission to the directory containing the Oracle inventory file. (See Section 9.4.4 for information about the location of the Oracle inventory directory.)

    The command line for the script has the following format:

    ```
    perl clone.pl ORACLE_HOME=OH_dir
                  ORACLE_HOME_NAME=OH_Name
                  -instance Instance_Name
                  [-db_vendor vendor_name
                  -db_user username
                  {-db_password db_pass | -db_obf_password db_obf_pass}
                  -db_string connect_db_string
                  -db_sid db_service_name ]
                  {-oc4jadmin_old_password old_admin_pass |
                   -oc4jadmin_obf_old_password old_obf_admin_pass}
                  {-oc4jadmin_new_password new_admin_pass |
                   -oc4jadmin_obf_new_password new_obf_admin_pass}
                  [-Ostring]
                  [-silent]
    ```

```
[-debug]
[-import]
[-help]
```

In the example, substitute the following for *perl*:

- On UNIX:

  ```
  $ORACLE_HOME/perl/bin/perl
  ```

- On Windows:

  ```
  %ORACLE_HOME%\perl\5.8.3\bin\MSWin32-x86-multi-thread\perl5.8.3
  ```

Table 9–2 describes the parameters and options for the `clone.pl` script.

Table 9–2   Parameters and Options for the clone.pl Script

Parameter or Option	Description
ORACLE_HOME	**Required.** The complete directory specification for the destination Oracle home. This parameter is required. If you do not supply this parameter, or if the value is invalid, the script exits.
	Do not use a trailing slash (UNIX) or backslash (Windows) when specifying the Oracle home.
ORACLE_HOME_NAME	**Required.** The name for the destination Oracle home (the Oracle home of the clone.)
-instance	**Required.** The instance name for the clone. The instance name must be different from the source instance and any other instances that use the same OracleAS Infrastructure or that are part of the same cluster topology.
-db_vendor	Required only in installations with an Oracle RAC database, Oracle BPEL Process Manager, or Oracle Enterprise Service Bus. The vendor for the database that stores the schemas for these products. The valid value is `oracle`.
-db_user	Required only in installations with an Oracle RAC database, Oracle BPEL Process Manager, or Oracle Enterprise Service Bus. The database user who created the schemas in the database. For example, SYS.
	If you do not supply a username, the script will exit.
-db_password	Required only in installations with an Oracle RAC database, Oracle BPEL Process Manager, or Oracle Enterprise Service Bus. The password for the database user. You can use this option or -db_obf_password.
	If you do not supply a username, the script will prompt you for one.
-db_obf_password	Required only in installations with an Oracle RAC database, Oracle BPEL Process Manager, or Oracle Enterprise Service Bus. The obfuscated password for the database user. You can use this option or -db_password.

Table 9–2   (Cont.) Parameters and Options for the clone.pl Script

Parameter or Option	Description
-db_string	Required only in installations with an Oracle RAC database, Oracle BPEL Process Manager, or Oracle Enterprise Service Bus.
	The format of the connect_db_string varies depending on the type of database
	For a single-instance database:
	hostname:port
	where hostname is the fully qualified name of the host containing the database and port is the port for the database listener.
	For a 10*g* Oracle RAC database:
	virtual_hostname_node1:port1^virtual_hostname_node2:port2[^...]
	where virtual_hostname_node1 is the virtual hostname for the first node, port1 is the port for the database listener on the first node. For each subsequent node, that is followed by a caret (^) and similar information.
	For a 9*i* Oracle RAC database:
	*hostname_node1:port1^hostname_node2:port2^*[...]
	where hostname_node1 is the hostname for the first node, port1 is the port for the database listener on the first node. For each subsequent node, that is followed by a caret (^) and similar information.
-db_sid	Required only in installations with an Oracle RAC database, Oracle BPEL Process Manager, or Oracle Enterprise Service Bus. The fully qualified service name for the database. (Note that this requires the service name, not the SID.) For example, orcl.us.oracle.com.
-oc4jadmin_old_password	**Required if -oc4jadmin_obf_old_password is not used.** The administrator oc4jadmin password for Oracle Application Server for the source instance. If you do not supply this option or -oc4jadmin_obf_old_password and the script is not running in silent mode, the script prompts the user for the password.
-oc4jadmin_obf_old_password	**Required if -oc4jadmin_old_password is not used.** The obfuscated administrator oc4jadmin password for Oracle Application Server for the source instance. If you do not supply this option or -oc4jadmin_old_password and the script is not running in silent mode, the script prompts the user for the password.
-oc4jadmin_new_password	**Required if -oc4jadmin_obf_new_password is not used.** A new password for the administrator oc4jadmin for Oracle Application Server for the cloned instance. If you do not supply this option or -oc4jadmin_obf_new_password and the script is not running in silent mode, the script prompts the user for the password.
	This password is used for the default OC4J instance, not for any other OC4J instances. See Section 9.5.3 for more information.

Table 9–2   (Cont.) Parameters and Options for the clone.pl Script

Parameter or Option	Description
-oc4jadmin_obf_new_password	**Required if -oc4jadmin_new_password is not used.** A new obfuscated password for the administrator oc4jadmin for Oracle Application Server for the cloned instance. If you do not supply this option or -oc4jadmin_new_password and the script is not running in silent mode, the script prompts the user for the password.
	This password is used for the default OC4J instance, not for any other OC4J instances. See Section 9.5.3 for more information.
-O	Specifies that any text following the option is passed to the Oracle Universal Installer command line. For example, you can use this option to pass the location of the oraparam.ini file to be used by Oracle Universal Installer, by using the following code:
	`'-O-paramFile C:\OraHome_1\oui\oraparam.ini'`
	Note that if the text you want to pass contains spaces or other delimiting characters, you must enclose the option in double quotation marks (").
	To pass multiple parameters to Oracle Universal Installer using this option, you can either pass all parameters with a single -O option or pass individual parameters using multiple -O options.
-silent	Runs the script in silent mode. If the command line does not contain the required password-related options, the script exits.
-debug	Runs the script in debug mode.
-import	Imports the page customizations and portlet metadata that are stored in MDS on the source instance to cloned instance. Also imports the portlet customizations (preferences). It imports the customizations from the .ear file generated by the -export option on the prepare_clone.pl command line. Use this option to migrate customizations that are associated to a WebCenter application from one location to another if the new instance will use a different MDS location.
	This option calls the import mode of the Oracle WebCenter Framework Predeployment tool. For more information about the Predeployment tool, see "Deploying Your WebCenter Application" in the *Oracle WebCenter Framework Developer's Guide*.
	The script imports the ear file created by the prepare_clone.pl -export option.
-help	Prints the usage for the script.

For example:

```
perl clone.pl ORACLE_HOME=/scratch/oracle/Ora_1013_B
              ORACLE_HOME_NAME=OH_1013_B
            -instance orcl_B
            -db_vendor oracle
            -db_user SYS
            -db_password my_db_pass
            -db_string my_host:1521
            -db_sid orcl.us.oracle.com
            -oc4jadmin_old_password my_old_admin_pass
```

```
-oc4jadmin_new_password my_new_admin_pass
'-O-paramFile /var/opt/oracle/oui/oraparam.ini'
-silent
```

5. If the source instance contains deployed WebCenter applications, the cloning script asks whether you want to use the same MDS location or a different location.

For example, if you are moving from a test environment to a production environment, you can specify a new MDS location. However, if you are expanding your environment by adding a new instance, Oracle recommends that you use the same MDS as the source instance.

```
Specified mds path is mds_path. Do you want to keep the original settings [n|y]
[y]:
```

To use the same MDS location, specify **y**.

To use a different MDS location, specify **n**. Then at the prompt, enter the new location. Note that you must specify an absolute path and that the location of the MDS must be accessible. That is, the user must have read and write permission.

If you specify an incorrect location, you can change the location after the cloning operation is complete. Change the location in the `adf-config.xml` file, which is located in the following directory:

```
ORACLE_HOME/j2ee/OC4J_Instance/applications/apps_name/adf/META-INF
```

In the directory specification, *OC4J_instance* is the name of the OC4J instance, and *apps_name* is the name of the applications.

6. If the source instance is a member of a multicast dynamic node discovery or static node discovery cluster, the script asks whether or not you want to keep the original cluster settings. See Section 9.4.5 for more details.

7. On UNIX, run the `root.sh` script in the Oracle home so that the cloned instance works properly. You must be logged in as the root user to run the script. The script is located in the cloned instance's Oracle home directory.

For example:

```
$ORACLE_HOME/root.sh
```

8. On UNIX, if this is the first Oracle installation on the computer, you must run the `orainstRoot.sh` script as the root user to register the Oracle Inventory directory. The script in located in the oraInventory directory.

The following file contains the location of the oraInventory directory:

```
ORACLE_HOME/clone/logs/clonetimestamp.log
```

9. If you are using a new MDS location, redeploy your applications on the cloned instance using Application Server Control. See "Deploying Your WebCenter Application" in the *Oracle WebCenter Framework Developer's Guide* for information on deploying WebCenter applications.

10. If the cloned instance is not using the same MDS location and it contains WebCenter applications, see "Using Cloning to Move from Stage to Production" in the *Oracle WebCenter Framework Developer's Guide* for additional steps.

11. If the instance contains Oracle BPEL Process Manager, redeploy all applications, as described in Section 9.5.5.

12. If the instance contains Oracle Enterprise Service Bus, import the services as described in Section 9.5.6.

13. Restart the cloned instance:

- On UNIX systems:

    ```
    ORACLE_HOME/opmn/bin/opmnctl stopall
    ORACLE_HOME/opmn/bin/opmnctl startall
    ```

- On Windows systems:

    ```
    ORACLE_HOME\opmn\bin\opmnctl stopall
    ORACLE_HOME\opmn\bin\opmnctl startall
    ```

Now, the cloned instance's configuration is identical to that of the source instance. Application Server Control and OPMN are able to start and stop all processes in the cloned instance, including any OC4J custom instances. All applications deployed should be visible and able to run as expected.

9.4.4 Locating and Viewing Log Files

The cloning script invokes multiple tools, each of which generates its own log files. However, the following log files, which are generated by Oracle Universal Installer and the cloning scripts, are the key ones of interest for diagnostic purposes:

- *Oracle_inventory*/logs/cloneActions*timestamp*.log: Contains a detailed log of the actions that occur during the Oracle Universal Installer part of the cloning.

- *Oracle_inventory*/logs/oraInstall*timestamp*.err: Contains information about errors that occur when Oracle Universal Installer is running.

- *Oracle_inventory*/logs/oraInstall*timestamp*.out: Contains other miscellaneous messages generated by Oracle Universal Installer.

- *Oracle_Home*/clone/logs/clone*timestamp*.log: Contains a detailed log of the actions that occur during the precloning and cloning operations.

- *Oracle_Home*/clone/logs/error*timestamp*.log: Contains information about errors that occur during the precloning and cloning operations. In addition, it contains all messages written to standard error (STDERR) by the multiple tools that are invoked by the cloning script. Depending upon the tool, some of these messages may be informational messages or error messages.

The format of the path is shown in UNIX format. For Windows, invert the slashes.

> **Note:** The following file contains the location of the Oracle Inventory directory:
>
> *Oracle_Home*/clone/logs/clone*timestamp*.log
>
> For example:
>
> ```
> Wed Jul  5 09:42:51 2006  INFO: Please check
> /scratch/oracleas/oraInventory/logs/cloneActions2006-07-05_
> 09-38-30AM.log for more details.
> ```
>
> On Windows systems, the location can be obtained from the registry:
> ```
> HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\INST_LOC
> ```

After the `clone.pl` script finishes running, consult these log files to get more information about the cloning process. To view the log files from Application Server Control, take the following steps:

1. Select **Logs** from the Home page.

2. In the View Logs page, select **ASClone** from the **Available Components** box. Click **Move** to move the selection to the **Selected Components** box.

3. Click **Search.**

    The log files are displayed in the **Results** table.

4. To view the log, click the log name in the **Log File** column.

9.4.5 Cloning Instances That Are Members of a Cluster Topology

You can clone a middle-tier instance that is a member of a cluster, but only if it is based on multicast dynamic node discovery or static node discovery. In these cases, the cloning script asks whether or not you want to keep the original cluster settings. For example:

```
cluster Config:<Multi Casting>detected for Source Instance:Do you want to keep
the original cluster settings(n|y)[y]:
```

If you respond `y`, the cloned instance will be part of the same cluster as the source instance.

If you respond `n`, the cloning script prompts you for a new IP address and port for the cluster. The type of cluster is preserved. For example, if the source instance is a member of a dynamic node discovery cluster and you respond `n`, you are prompted to enter a new multicast discovery IP address and port. You cannot change the cluster type to a different type, such as a static node discovery cluster, during the cloning procedure.

You cannot clone an instance that is a member of a cluster based on a cross-topology gateway or manual node-to-node configuration. You must remove them from the cluster first.

9.5 Considerations and Limitations for Cloning

The following sections provide details of considerations and limitations affecting cloning in general and specific components in the cloned Oracle home:

*   General Considerations and Limitations for Cloning

*   Considerations for Cloning Oracle HTTP Server

*   Considerations for Cloning Oracle Containers for J2EE (OC4J)

*   Considerations for Cloning Application Server Control

*   Considerations for Cloning Oracle BPEL Process Manager

*   Considerations for Cloning Oracle Enterprise Service Bus

*   Considerations for Cloning Oracle WebCenter Framework

9.5.1 General Considerations and Limitations for Cloning

For this release, you cannot clone the following:

- OracleAS Infrastructure components (Oracle Identity Management and OracleAS Metadata Repository)

- Oracle Content DB

- Oracle TopLink

- An instance that is connected to Oracle Database Lite

- Oracle Web Services Manager. The cloning operation copies the files from the source to the cloned instance, but you cannot use the cloned Oracle Web Services Manager instance.

Note the following important additional considerations about cloning:

- Oracle recommends that you stop the source instance completely before you run the `prepare_clone.pl` script.

- You can clone an instance to the same operating system.

- The cloned instance will use the hostname of the machine, rather than the virtual hostname, even if you set the VIRTUAL_HOST_NAME environment variable during installation of the source instance and before performing the cloning operation.

- If you set VIRTUAL_HOST_NAME when you installed the Oracle Application Server instance, you must update the OUI_HOSTNAME parameter in the following file with the VIRTUAL_HOST_NAME, before you run the prepare_clone script:

  ```
  UNIX) ORACLE_HOME/inventory/Clone/clone.xml
  (Windows) ORACLE_HOME\inventory\Clone\clone.xml
  ```

- You may need to update the security certificates to match the new host name and to set up the certificates. See Chapter 11 for information about managing wallets and certificates.

- You can clone an Oracle Application Server middle tier that is connected to Oracle Identity Management. However, the cloned middle-tier instance will not be associated with Oracle Identity Management. You must manually associate the cloned middle tier instance to Oracle Identity Management. To associate the middle tier with Oracle Identity Management, see Section 6.7, "Configuring Instances to Use 10.1.4 or 10.1.2 Oracle Identity Management" for instructions.

- If you have changed the default file permissions for configuration files, those file permissions are not preserved by cloning.

- User customizations for the following components are not preserved. The status of these components are reset to the default:

  - Oracle Application Development Framework

  - Port tunneling

  - UIX

  - XDK

- Cloning does not carry over all the dependencies of the source Oracle home, such as loadable modules or application-specific libraries to the cloned home, because cloning proceeds by copying the entire source Oracle home to the destination Oracle home. Any files outside the source Oracle home are not automatically copied. Hence, any applications that refer to files outside the source Oracle home may not work properly in the cloned home.

You may need to copy the files manually to the destination host after extracting the archived source Oracle home, but before running the clone.pl script.

- If you created symbolic links to files or applications outside the source Oracle home (for example, to Oracle Wallet files that are not stored in the default location), you must re-create the link manually in the cloned home for your applications to work properly.

- If ORACLE_HOME/portal/conf/javacache.xml is present in the source environment, you must manually modify this file in the cloned environment.

  In this file, change the following:

  - In the <location> element, replace the value of the source ORACLE_HOME with the value of the cloned ORACLE_HOME.

  - In the <coordinator discovery-port> element, change the value to the new port number.

- The cloning operation generates default ports for the cloned instance. To specify other ports, you can use the staticports.ini file as described in Section 9.6.2. If you specify ports less than 1024 on UNIX, the cloned instance will not start during the cloning operation. After the cloning process is completed, you must run the root.sh script with root privileges, then start the processes.

- The cloning process does not configure a Load Balancing Router to recognize the cloned instance. If you use a Load Balancing Router in your environment, you must manually configure the Load Balancing Router, including any invalidation port.

- If a cloning operation fails, but it results in the Oracle home being registered with Oracle Inventory, you cannot use the same Oracle home in subsequent cloning operations. Either use another directory and name for the Oracle home in subsequent cloning operations or deinstall the Oracle home before attempting another cloning operation.

9.5.2 Considerations for Cloning Oracle HTTP Server

The following describes important information about cloning Oracle HTTP Server:

- All configuration information in the following files is updated:

  - *Oracle_Home*/Apache/Apache/conf/httpd.conf

  - *Oracle_Home*/Apache/Apache/conf/mod_oc4j.conf

  - *Oracle_Home*/Apache/Apache/conf/oracle_apache.conf

  - *Oracle_Home*/Apache/modplsql/conf/cache.conf

  - *Oracle_Home*/Apache/modplsql/conf/dads.conf

  - *Oracle_Home*/Apache/modplsql/conf/plsql.conf

  - *Oracle_Home*/Apache/oradav/conf/moddav.conf

  The format of the paths are shown in UNIX format. For Windows, invert the slashes.

  The cloning script preserves the source settings and updates these files with new environment parameters.

  Note that cloning only updates the files it knows about, that is, files that are part of the original installation. In particular, cloning does not update configuration files that the user added to the "include" list in files such as httpd.conf, oracle_

apache.conf, dads.conf, plsql.conf, olap.conf, or moddav.conf. You can, however, explicitly add the "include" files to the list of files that cloning will update. See Section 9.6.3 for details on how to update custom settings.

- Cloning preserves all VirtualHost directives in httpd.conf. It replaces any references to the source home inside these directives. However, cloning does not change the IP numbers or port numbers that these virtual hosts listen to.

  If these values are not valid for the destination environment, then you must do one of the following:

  - Register these changes with the clone script to be updated during cloning. See Section 9.6.3 for more information.

  - Update them manually in httpd.conf after cloning.

- If you changed the port number in httpd.conf to use the Load Balancing Router port rather than the local Oracle HTTP Server port, that change is lost after cloning. You must edit the httpd.conf file in the cloned home to change the port number to the Load Balancing Router port.

- Cloning is not supported if you are using Oracle HTTP Server based on Apache 1.3 or Apache 2.0. (These are not installed by default, but are included in the companion CD-ROM.)

9.5.3 Considerations for Cloning Oracle Containers for J2EE (OC4J)

This following lists considerations in cloning OC4J:

- On the source Oracle home during the prepare phase of the cloning process, do not attempt to undeploy OC4J applications or perform other administrative work for OC4J applications while the prepare_clone.pl script is running.

- You must manually register files that contain environment-specific information in custom OC4J instances so that those files are updated during cloning. An example of such a file is oc4j.properties. See Section 9.3 for details.

- If the OC4J instance uses an Oracle HTTP Server instance that is not part of the source Oracle home, the cloning procedure does not update the mod_oc4j.conf file for the Oracle HTTP Server. You must manually add the instance to the mod_oc4j.conf file.

- During cloning, you specify a password for the default OC4J instance. If the source instance contains OC4J instances other than the default, the OC4J instances in the cloned instance will use the same password as those in the source. That is, they will not use the password specified for the default OC4J instance. You can change the password using Application Server Control.

  Note that each OC4J instance in a group must have the same oc4jadmin password. See Section 2.3.3.2 for more information.

- OPMN can manage all cloned default and custom OC4J instances.

- Grid Control Console on the cloned instance can manage default and custom OC4J instances.

The following describes which OC4J components are preserved:

- All default OC4J instances are preserved.

- Custom OC4J instances that you created, as well as applications deployed in them, are preserved. However, external dependencies for these applications that are not in the Oracle home are not copied to the cloned home and will be lost.

- Data source information in `data-sources.xml` is preserved.

- User configurations in `jms.xml`, `java2.policy`, `jazn.xml`, `jazn-data.xml`, `global-web-application.xml`, and `application.xml` are preserved.

9.5.4 Considerations for Cloning Application Server Control

Note the following considerations for cloning Application Server Control:

- If the source instance contains its own Application Server Control, the cloned instance will contain its own Application Server Control, which will manage the cloned instance.

- If the source instance is managed by an Application Server Control that is deployed in a different Oracle home, the cloned instance will be managed by the same Application Server Control.

The SSL settings of the source instance are preserved, by preserving the `default-web-site.xml` file. In other words, if the source Application Server Control was configured for HTTPS, then the cloned Application Server Control will be as well.

9.5.5 Considerations for Cloning Oracle BPEL Process Manager

If you clone an instance that contains Oracle BPEL Process Manager, you must take the following steps after cloning:

1. Stop the Oracle BPEL Process Manager server:

   ```
   (UNIX) ORACLE_HOME/bpel/bin/shutdownorabpel.sh
   (Windows) ORACLE_HOME\bpel\bin\shutdownorabpel.bat
   ```

2. Ensure that the tmp directory, if it is exists in the cloned instance, is removed. The directory is located at:

   ```
   (UNIX) ORACLE_HOME/bpel/domains/domainname/tmp/*
   (Windows) ORACLE_HOME\bpel\domains\domainname\tmp\*
   ```

3. Restart the Oracle BPEL Process Manager server:

   ```
   (UNIX) ORACLE_HOME/bpel/bin/startorabpel.sh
   (Windows) ORACLE_HOME\bpel\bin\startorabpel.bat
   ```

4. Redeploy all applications. See the *Oracle BPEL Process Manager Administrator's Guide*.

9.5.6 Considerations for Cloning Oracle Enterprise Service Bus

If you clone an instance that contains Oracle Enterprise Service Bus, you must export the Oracle Enterprise Service Bus services from the source instance before you clone the instance and you must import those services to the cloned instance after cloning:

- On the source instance before cloning, take the following steps to export the services:

  1. Run the following script, which sets the Oracle home and system variables:

     ```
     (UNIX) ORACLE_HOME/integration/esb/bin/esbdevprompt.sh
     (Windows) ORACLE_HOME\integration\esb\bin\esbdevprompt.bat
     ```

  2. In the command-line window from which you ran the script, run the following command to export the Oracle Enterprise Service Bus services:

```
ant export-params
-Dparamfile ORACLE_HOME\integration\DateTimeStamp\esbparam.properties
-DDB_URL=jdbc_connectString:@//hostname:port/db_service_name
-DDB_USER=oraesb -DDB_PASSWORD=esb_pwd
-DDB_DRIVER=oracle.jdbc.driver.OracleDriver
```

In the example, *hostname* is the host containing the repository for Oracle Enterprise Service Bus, *port* is the port for the repository, *db_service_name* is the database service name, and *esb_pwd* is the password for the ORAESB user. For example, on Windows:

```
ant export-params
-Dparamfile ORACLE_HOME\integration\20060828_1503\esbparam.properties
-DDB_URL=jdbc:oracle:thin:@//sta.oracle.com:1521/orcl.us.oracle.com
-DDB_USER=oraesb -DDB_PASSWORD=ORAESB123
-DDB_DRIVER=oracle.jdbc.driver.OracleDriver
```

- On the cloned instance, after you have run the cloning script, take the following steps to import the services:

  1. Copy the `esbparam.properties` file from the source instance to the following location on the cloned instance:

     ```
     (UNIX) ORACLE_HOME\integration\DateTimeStamp\esbparam.properties
     (Windows ORACLE_HOME/integration/DateTimeStamp/esbparam.properties
     ```

  2. Edit the `esbparam.properties` file on the cloned instance to reflect the new environment.

     If the cloned instance is on the same host as the source instance, change the DT_OC4J_HTTP_PORT to the new port number. For example, if the HTTP listen port is 7778, the value of the property would be:

     ```
     DT_OC4J_HTTP_PORT=7778
     ```

     If the cloned instance is on a different host than the source, change the DT_OC4J_HOST property to the new hostname and change the DT_OC4J_HTTP_PORT to the new port number. For example, if the hostname of the cloned instance is `newhost.oracle.com` and the HTTP listen port is 7778, the value of the properties would be:

     ```
     DT_OC4J_HOST=newhost.oracle.com
     DT_OC4J_HTTP_PORT=7778
     ```

  3. In a command-line window, run the following script, which sets the Oracle home and system variables:

     ```
     (UNIX) ORACLE_HOME/integration/esb/bin/esbdevprompt.sh
     (Windows) ORACLE_HOME\integration\esb\bin\esbdevprompt.bat
     ```

  4. Run the following command from the same command-line window as in Step 3:

     ```
     ant import-params
     -Dparamfile ORACLE_HOME\integration\DateTimeStamp\esbparam.properties
     -DDB_URL=jdbc_connectString:@//hostname:port/db_service_name
     -DDB_USER=oraesb -DDB_PASSWORD=esb_pwd
     -DDB_DRIVER=oracle.jdbc.driver.OracleDriver
     ```

     In the example, *hostname* is the host containing the repository for Oracle Enterprise Service Bus, *port* is the port for the repository, *db_service_name* is the

database service name, and *esb_pwd* is the password for the ORAESB user. For example, on Windows:

```
ant import-params
-Dparamfile ORACLE_HOME\integration\20060828_1503\esbparam.properties
-DDB_URL=jdbc:oracle:thin:@//sta1.oracle.com:1521/orcl.us.oracle.com
-DDB_USER=oraesb -DDB_PASSWORD=ORAESB123
-DDB_DRIVER=oracle.jdbc.driver.OracleDriver
```

5. Redeploy all applications, including Oracle BPEL Process Manager applications.

9.5.7 Considerations for Cloning Oracle WebCenter Framework

Note the following considerations for cloning Oracle WebCenter Framework:

- If the Oracle Metadata Services (MDS) location used by WebCenter applications is not located within the Oracle home, the cloned instance cannot use the MDS location unless it is on a shared drive.

- If the source instance contains deployed WebCenter applications, the cloning script asks whether you want to use the same MDS location or a different location.

  For example, if you are moving from a test environment to a production environment, you can specify a new MDS location. However, if you are expanding your environment by adding a new instance, Oracle recommends that you use the same MDS location as the source instance.

- To copy customizations, such as those made to the Producers of a deployed application, on the source instance to a new MDS location, use the `-export` option on the `prepare_clone` command line and the `-import` option on the `clone` command line. See Section 9.4.3.

For more information about cloning and Oracle WebCenter Framework, see the section "Cloning WebCenter Applications" in the *Oracle WebCenter Framework Developer's Guide*.

9.6 Customizing the Cloning Process

The default cloning process is adequate for most cases. However, you can customize some aspects of the cloning process by performing manual configuration steps, as described in the following sections:

- Specifying Oracle Universal Installer Parameters
- Assigning Custom Ports
- Updating Custom Data

9.6.1 Specifying Oracle Universal Installer Parameters

When cloning an instance, you do not directly invoke Oracle Universal Installer. However, you can still pass information to Oracle Universal Installer indirectly, by specifying any Oracle Universal Installer parameters that you normally specify on the command line in the configuration file `cs.properties`. This file is located in the following directory:

```
(UNIX) ORACLE_HOME/clone/ias/config
(Windows) ORACLE_HOME\clone\ias\config
```

For example, to specify a nondefault location for the Oracle inventory file on UNIX, you can add the following line to the `cs.properties` file:

```
clone_command_line= -invptrloc /private/oracle/oraInst.loc
```

To specify multiple arguments, append the argument to the `clone_command_line`, separating each argument with a space. Do not add additional `clone_command_line` lines. The following example shows how to specify two arguments, on Linux:

```
clone_command_line= -silent -invptrloc /private/oracle/oraInst.loc
oracle.as.j2ee.top:szl_PortListSelect="{YES,/tmp/staticports.ini}"
```

In addition, you can specify information to be passed to the Oracle Universal Installer command line by using the `-O`*`string`* option. For example, you can use this option to pass the location of the `oraparam.ini` file to be used by Oracle Universal Installer, by using the following code:

```
'-O-paramFile C:\OraHome_1\oui\oraparam.ini'
```

9.6.2 Assigning Custom Ports

By default, the cloning script automatically assigns free ports to the components. The algorithm for assigning default ports when cloning is the same as that used when installing Oracle Application Server.

When installing a new Oracle Application Server instance, you can specify the ports to be used by listing them in a `staticports.ini` file. Then, this file is passed as the value of a parameter when calling Oracle Universal Installer. For more information on how ports are assigned and on using the `staticports.ini` file, please refer to the *Oracle Application Server Installation Guide* for your platform.

When cloning an instance, you do not directly invoke Oracle Universal Installer. Hence, you cannot assign custom ports by specifying a `staticports.ini` file on the command line. However, you can still pass port information to Oracle Universal Installer indirectly, by specifying the location of the `staticports.ini` file in the following configuration file:

```
(UNIX) ORACLE_HOME/clone/ias/config/cs.properties
(Windows) ORACLE_HOME\clone\ias\config\cs.properties
```

For example, if you want to use ports less than 1024, you can specify them in the `staticports.ini` file and specify the location of the `staticports.ini` file in the `cs.properties` file.

To assign custom ports during cloning:

1. List the port numbers in the `staticports.ini` file, as explained in the *Oracle Application Server Installation Guide* for your platform.

2. Specify the location of the `staticports.ini` file by adding the information to the `clone_command_line` in the `cs.properties` file. For example, on Linux:

    ```
    clone_command_line= oracle.as.j2ee.top:szl_PortListSelect="{YES,
    /tmp/staticports.ini}"
    ```

The ports listed in the `staticports.ini` file are read during cloning, and Oracle Universal Installer assigns the port numbers accordingly.

If you specify ports less than 1024 on UNIX, the cloned instance will not start during the cloning operation. After the cloning process is completed, you must run the `root.sh` script with root privileges, then start the processes.

> **Note:** By default, Oracle Universal Installer saves all user input at installation and uses it to automate the actions when cloning. As a result, if you used a `staticports.ini` file to install the source instance, Oracle Universal Installer will, by default, use the same `staticports.ini` file. This happens even if you do not specify a `staticports.ini` file when you clone the instance. To override this behavior and let Oracle Universal Installer generate new ports, use the following line to the `cs.properties` file:
>
> ```
> oracle.as.j2ee.top:szl_PortListSelect="{\"NO\", \"\"}"
> ```

9.6.3 Updating Custom Data

By default, the cloning scripts update key configuration files in the Oracle home so that they contain the correct information for the destination environment. Section 9.4.3 contains a partial listing of the files that are updated.

You can modify the default cloning process to update custom data that is not updated by default. Information about which files to update during cloning and which entries to update in those files is contained in another set of files, which are read by the cloning scripts. By editing these files, you can:

- Preserve changes you have made to files present in the source Oracle home that are not updated by default during cloning

- Preserve changes you have made to files that are updated by default during cloning, but which are not normally preserved by the cloning process

These changes are made by a Java utility called FileFixer, which searches for specific text strings in a file by matching regular expressions, and updates them to their new values. Note that FileFixer searches for patterns one line at a time. It cannot match patterns across lines.

The changes that you can make include the following:

- Change the host name in a file.

  To do this, for the file in which the host name needs to be changed, add the path name for the file, relative to the Oracle home, to the following file:

  ```
  (UNIX) ORACLE_HOME/chgip/config/hostname.lst
  (Windows) ORACLE_HOME\chgip\config\hostname.lst
  ```

- Update all occurrences of the Oracle home in a file from the old to the new value.

  To do this, add a `replace` element tag in the XML configuration file, `fixup_script.xml.tmpl`. This file is located in the following directory:

  ```
  (UNIX) ORACLE_HOME/clone/ias/config
  (Windows) ORACLE_HOME\clone\ias\config
  ```

  The value of the file_name attribute specifies the name and location of the file in which the replacement should occur. For example, the following tag updates the Oracle home value in the file `server.xml`.

  ```
  <cfw:operation>
    <replace file_name="%NEW_HOME%/j2ee/home/config/server.xml">
        <cfw:replaceCommand>
            <cfw:pattern>(%OLD_HOME%)</cfw:pattern>
            <cfw:value_ref>1</cfw:value_ref>
            <cfw:new_value>%NEW_HOME%</cfw:new_value>
  ```

```
                    </cfw:replaceCommand>
                </replace>
            </cfw:operation>
```

9.7 Example: Using Cloning to Expand an Oracle Application Server Cluster

A common use of cloning is to expand the size of an Oracle Application Server cluster topology. Consider a cluster consisting of multiple Oracle WebCenter Framework and Oracle HTTP Server middle tiers, with identical configuration and application deployment. To expand the cluster, you want to create a new middle-tier instance that has the same configuration as the other instances and is part of the same cluster.

This example assumes that:

- The source instance is a member of a cluster based on multicast dynamic node discovery. The cloned instance will be part of the same cluster as the source instance.

- The source instance contains a file-based MDS that is located in the Oracle home and is on an NFS-shared disk.

Note that you cannot clone Oracle Web Services Manager. See Section 9.2 for more information

The strategy for expanding a cluster topology consists of the following steps:

1. Prepare the source instance for cloning, using the steps described in Section 9.4.2:

   a. Perform Step 2, as described in Section 9.4.2.

   b. Perform Step 3 in Section 9.4.2.

   In that step, use the -export option to export customizations made to the producers of a deployed application on the source instance to an .ear file. For example:

   ```
   perl prepare_clone.pl ORACLE_HOME=/scratch/oracleas/Ora_10132 -export
   ```

   c. Perform Step 4 as described in Section 9.4.2.

2. Clone the source instance to create a new instance, using the steps described in Section 9.4.3:

   a. Perform Steps 1, 2, and 3 as described in Section 9.4.3.

   b. Perform Step 4 in Section 9.4.3.

   In that step, you must change the name of the cloned instance, by specifying it on the command line. In addition, you must specify the -import option to import the .ear file generated by the prepare_clone operation. For example:

   ```
   perl clone.pl ORACLE_HOME=/scratch/oracle/Ora_10132_B
                 ORACLE_HOME_NAME=OH_10132B
                 -instance WebC
                 -oc4jadmin_old_password my_old_admin_pass
                 -oc4jadmin_new_password my_new_admin_pass
                 -import
   ```

   In this example, the instance name for the cloned instance is WebC.

3. Perform Steps 6 through 8 in Section 9.4.3, if necessary.

4. Perform Steps 9 through 10 in Section 9.4.3, if necessary.

5. Perform Step 13 in Section 9.4.3.

Part IV

Secure Sockets Layer (SSL)

This part contains the following chapters:

10

Overview of Secure Sockets Layer (SSL) in Oracle Application Server

In Oracle Application Server, components send requests to and receive responses from other components. These components can be Oracle Application Server components (such as Oracle HTTP Server, OC4J applications, or OracleAS Single Sign-On) or external clients such as browsers.

> **Note:** In this chapter, references to any of the following Oracle Application Server products are applicable for Release 10.1.4, Release 2 (10.1.2) or earlier software only:
>
> - OracleAS Single Sign-On
> - OracleAS Web Cache
> - OracleAS Certificate Authority
> - Oracle Identity Management
> - OracleAS Portal

To secure these communications, you can configure Oracle Application Server to use SSL, which is an industry standard for securing communications. Oracle Application Server supports SSL versions 2 and 3, as well as TLS version 1.

This chapter provides an overview of SSL and how you can use it with Oracle Application Server. It contains the following topics:

- What SSL Provides
- About Private and Public Key Cryptography
- How an SSL Session Is Set Up (the "SSL Handshake")
- Requirements for Using SSL in Oracle Application Server
- Certificates and Oracle Wallets
- SSL Configuration Overview
- Integration with Hardware Security Modules

10.1 What SSL Provides

SSL secures communication by providing message encryption, integrity, and authentication. The SSL standard allows the involved components (such as browsers

and HTTP servers) to negotiate which encryption, authentication, and integrity mechanisms to use.

- Encryption allows only the intended recipient to read the message. SSL can use different encryption algorithms to encrypt messages. During the SSL handshake that occurs at the start of each SSL session, the client and the server negotiate which algorithm to use. Examples of encryption algorithms supported by SSL include AES, RC4, and 3DES.

- Integrity ensures that a message sent by a client is received intact by the server, untampered. To ensure message integrity, the client hashes the message into a digest using a hash function and sends this **message digest** to the server. The server also hashes the message into a digest and compares the digests. Because SSL uses hash functions that make it computationally infeasible to produce the same digest from two different messages, the server can tell that if the digests do not match, then someone had tampered with the message. An example of a hash function supported by SSL is SHA1.

- Authentication enables the server and client to check that the other party is who it claims to be. When a client initiates an SSL session, the server typically sends its certificate to the client. Certificates are digital identities that are issued by trusted certificate authorities, such as Verisign. Section 10.5 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.

10.2 About Private and Public Key Cryptography

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

Private Key Cryptography

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

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

Public Key Cryptography

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

Public key algorithms can guarantee the secrecy of a message, but they do not necessarily guarantee secure communication because they do not verify the identities of the communicating parties. To establish secure communication, it is important to verify that the public key used to encrypt a message does in fact belong to the target recipient. Otherwise, a third party can potentially eavesdrop on the communication

and intercept public key requests, substituting its own public key for a legitimate key (the **man-in-the-middle** attack).

To avoid such an attack, it is necessary to verify the owner of the public key, a process called authentication. Authentication can be accomplished through a certificate authority (CA), which is a third party trusted by both of the communicating parties.

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

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

10.3 How an SSL Session Is Set Up (the "SSL Handshake")

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 10–1 shows the handshake messages for a typical SSL connection between a Web server and a browser. The following steps are shown in the figure:

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

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

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

   - The algorithm to use. The server selected this from the list sent by the client.

   - A random number, which will be used to generate the keys.

3. The server sends its certificate to the client.

4. The client authenticates the server using the server's certificate.

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

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

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

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

   - Encryption key that the client uses to encrypt data before sending it to the server

- Encryption key that the server uses to encrypt data before sending it to the client

- Key that the client uses to create a message digest of the data

- Key that the server uses to create a message digest of the data

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

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

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

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

Figure 10–1   SSL Handshake
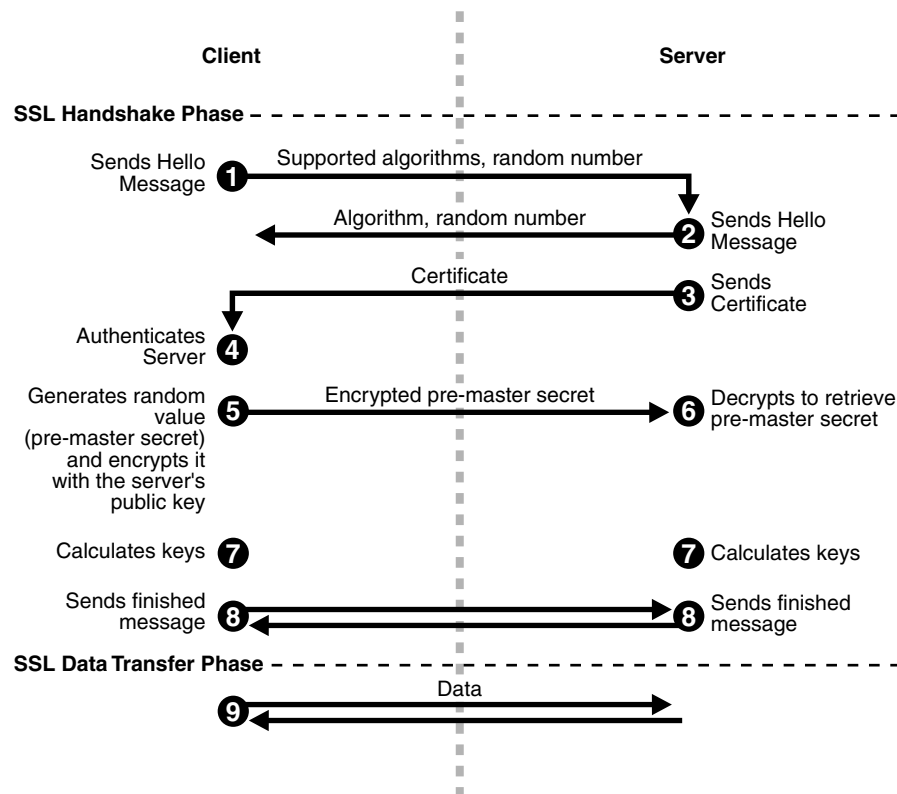
10.4 Requirements for Using SSL in Oracle Application Server

To use SSL in Oracle Application Server:

- You need a certificate and an Oracle wallet for your site. This certificate is used by clients to verify that they are not connecting to an imposter site.

- If you need to authenticate your clients, the clients will need certificates as well.

- You need to configure Oracle Application Server components (for example, Oracle HTTP Server) to accept and transmit messages over SSL.

- SSL is resource-intensive. If you expect heavy SSL traffic, then you should consider getting an SSL accelerator.

The following sections in this chapter describe these topics in more detail.

10.5 Certificates and Oracle Wallets

You need a certificate for your site. When clients connect to your site requesting SSL communication, you usually have to send your certificate to them so that they can authenticate you.

Oracle Application Server supports X.509 V3 certificates, and certificates that conform to the PKIX standard (RFC 3280).

10.5.1 How to Get a Certificate

You get certificates from certificate authorities (CAs). CAs are trusted entities who sign the certificates that they issue with their private key. Clients can verify the issuer of a certificate (by using the CA's public key). Examples of CAs include Verisign (http://www.verisign.com) and Thawte (http://www.thawte.com).

Oracle Application Server also has a certificate authority, called OracleAS Certificate Authority (OCA). You can use it to set up your own certificate authority. See the *Oracle Application Server Certificate Authority Administrator's Guide* for details.

To get a certificate, you submit a certificate request to a CA. The certificate request contains your information including your public key. You can use tools to generate a certificate request; these tools can generate private and public key pairs for you. Examples of tools that can generate certificate requests include Oracle Wallet Manager and Sun's keytool (for OC4J only). For information on Oracle Wallet Manager, see Chapter 11, "Managing Wallets and Certificates".

Among other items, a certificate includes the following pieces of data:

- Certificate owner's name

- Certificate owner's public key

- CA's name

- Certificate expiration date

- Certificate serial number

Certificates are valid until they expire or until they are revoked.

Note that if you use OracleAS Certificate Authority (OCA) to create certificates for your server, most browsers will not accept these certificates without input from the browser user. This is because most browsers are preconfigured to accept certificates from certain CAs, and OCA is not one of them. The browser will reject the certificate from the server unless the user chooses to accept certificates from the server or import the CA's certificate.

This problem exists for all CAs until the CA's certificate is imported into the browsers. For more information, see the *Oracle Application Server Certificate Authority Administrator's Guide*.

10.5.2 Oracle Wallet

An Oracle wallet is a container that stores your credentials, such as 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 are password-protected.

You manage Oracle wallets using Oracle Wallet Manager. Use Oracle Wallet Manager to perform tasks such as creating Oracle wallets, creating certificate requests, importing certificates into the wallet, and uploading wallets to an LDAP directory.

Oracle Wallet Manager supports PKCS #11 and PKCS #12 wallets:

- Scenario 1: You generate a certificate request using Oracle Wallet Manager and decide to store the private key on the file system. When you get your certificate from the CA, you can import it into an Oracle wallet. This wallet uses the PKCS #12 format. See Section 11.1.4.2.1 for details.

- Scenario 2: You generate a certificate request using Oracle Wallet Manager and decide to store the private key on a hardware security module. When you get your certificate from the CA, you can import it into an Oracle wallet. This wallet uses the PKCS #11 format. See Section 11.1.4.2.2 for details.

- Scenario 3: You already have a certificate in a wallet that uses the PKCS #12 format, and you want to use it with Oracle Application Server. The wallet was created using a third-party tool. In this case, use the tool that was used to create the wallet and export the wallet to a file on the file system. You can then import the wallet. See Section 11.1.5.1.3 for details.

Components That Use Oracle Wallets

Oracle Application Server components that act as SSL servers need Oracle wallets (the wallet already contains the certificate that you want the server to use). Examples of these components include Oracle HTTP Server, OracleAS Web Cache, OPMN, Oracle Internet Directory, and the Port Tunneling daemon (`iaspt`).

You configure the component with the location of the Oracle wallet. For example, to configure Oracle HTTP Server for SSL, you specify the location of the wallet using the `SSLWallet` directive. Refer to the component guide for specific instructions on how to specify the wallet location for the component.

> **Note:** The OC4J component uses a *keystore* instead of an Oracle wallet to store its certificate. You use a tool called `keytool` to import certificates into keystores. See the *Oracle Containers for J2EE Security Guide* for details on keystores and `keytool`.

10.5.3 Client Certificates

If you need to authenticate your clients, you can configure the Oracle HTTP Server to require clients to send their certificates. Clients can also get their certificates from CAs.

If the clients are Oracle components, for example, OracleAS Web Cache can act as a client when communicating with Oracle HTTP Server, the client component can store its certificate in an Oracle wallet. OPMN also acts as a client when configured for SSL.

If the client is a browser, the client does not need an Oracle wallet. You can just import the certificate into the browser.

Other types of clients, such as SOAP or Web Services clients, have their own ways of configuring certificates and certificate stores.

10.6 SSL Configuration Overview

You enable components in Oracle Application Server to use SSL using Application Server Control. In some cases, you edit configuration files by hand.

SSL secures communication between two parties: a client and a server. If three or more parties are involved, for example, client browser, OracleAS Web Cache, Oracle HTTP Server, and OC4J, then you may have to configure all components to use SSL.

Figure 10–2 shows typical communication paths between Oracle Application Server components and the protocols that they use. For example, browsers use HTTP to communicate with OracleAS Web Cache, and Oracle HTTP Server uses AJP to communicate with OC4J. All these protocols can work with SSL.

Figure 10–2   Communication Paths Between Components in Oracle Application Server
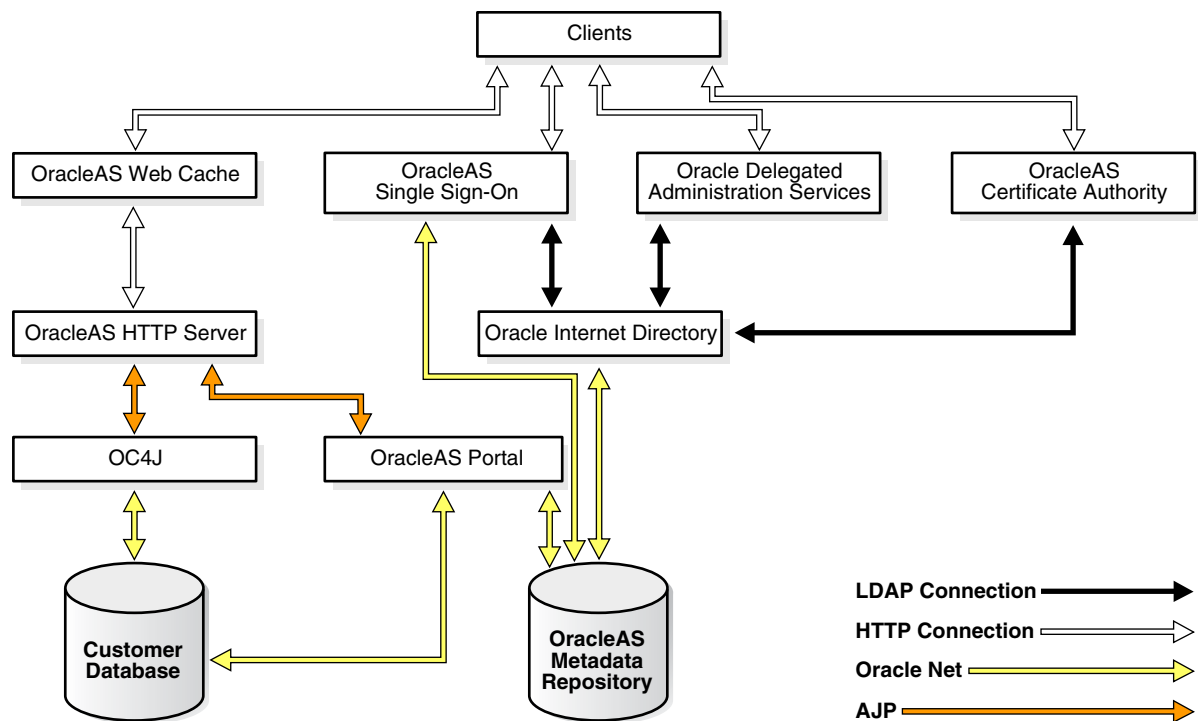
10.6.1 Default SSL Configuration

If you select the default options in the Oracle Application Server installation, none of the components are configured for SSL.

On installation screens where you specify the Oracle Internet Directory host and port, there is an option marked "Use only SSL connections with this Oracle Internet Directory." If you select this option, you must provide Oracle Internet Directory's SSL port number, and the installer configures the components to use SSL only to communicate with Oracle Internet Directory during run-time.

10.6.2 Partial SSL Configuration

Oracle Application Server enables you to configure SSL for only the paths you want to secure. There are many paths used by components, as shown in Figure 10–2.

You might not want to secure all paths for the following reasons:

- SSL is resource-intensive. If you have heavy SSL traffic, then you probably need to off load SSL processing to an SSL accelerator. See Section 10.7.

- If your computers are behind firewalls, you might need to secure only paths that are accessed by the public. For example, you might need to secure only OracleAS Web Cache and Oracle HTTP Server if the public can access only these components.

10.7 Integration with Hardware Security Modules

When clients connect to your site using SSL, the extra processing required for SSL strains your servers, and your site as a whole (SSL as well as non-SSL connections) will experience slower performance and throughput. You should consider using SSL accelerator hardware to off load SSL computations and improve performance.

Types of SSL accelerators:

- Section 10.7.1, "Protocol Converters"

- Section 10.7.2, "Mathematics Accelerators (PKCS #11 Integration)"

10.7.1 Protocol Converters

Protocol converters convert HTTPS traffic to HTTP. Protocol converters are standalone hardware machines. Oracle Application Server supports protocol converters from companies such as:

- F5 (`http://www.f5.com`)

- Cisco (`http://www.cisco.com`)

- SonicWall (`http://www.sonicwall.com`)

> **Note:** SSL connections to protocol converters terminate at the protocol converter. When the converters forward the requests to Oracle Application Server, most of them do so in an **unencrypted** fashion.
>
> For the protocol converters that forward the requests to Oracle Application Server using SSL, this is still faster than not using a protocol converter because using a protocol converter eliminates most SSL key exchanges (which is the expensive operation).

10.7.2 Mathematics Accelerators (PKCS #11 Integration)

Mathematics accelerators improve the speed of math operations used by SSL. Such devices are usually plugged into a server (often through TCP/IP). Such devices often have additional features such as key management and secure key stores.

Oracle Application Server supports mathematics accelerators that follow the PKCS #11 standard. For a list of certified accelerators, you can check the Oracle*MetaLink* site, `http://www.oracle.com/support/metalink/index.html`.

11

Managing Wallets and Certificates

This chapter explains how to obtain and manage security credentials for Oracle Application Server resources. Security administrators can use Oracle Wallet Manager and its command-line utility, `orapki,` to manage public key infrastructure (PKI) credentials on Oracle clients and servers. These tools create credentials that can be read by Oracle Database, Oracle Application Server, and the Oracle Identity Management infrastructure.

> **Note:** In this chapter, references to any of the following Oracle Application Server products are applicable for Release 10.1.4, Release 2 (10.1.2), or earlier software only:
> - Oracle Identity Management
> - Oracle Internet Directory

This chapter contains the following topics:

- Using Oracle Wallet Manager
- Performing Certificate Validation and CRL Management with the orapki Utility
- Interoperability with X.509 Certificates

> **Note:** If you already have certificates provisioned, the following sections may provide all of the information you need:
> Section 11.1.2, "Starting Oracle Wallet Manager"
> Section 11.3, "Interoperability with X.509 Certificates"

11.1 Using Oracle Wallet Manager

This section describes Oracle Wallet Manager, a graphical user interface tool used to manage PKI certificates. It contains the following topics:

- Oracle Wallet Manager Overview
- Starting Oracle Wallet Manager
- How to Create a Complete Wallet: Process Overview
- Managing Wallets
- Managing Certificates

11.1.1 Oracle Wallet Manager Overview

Oracle Wallet Manager is an application used to manage and edit security credentials in Oracle wallets. A wallet is a password-protected container that stores authentication and signing credentials, including private keys, certificates, and trusted certificates, all of which are used by SSL for strong authentication. You can use Oracle Wallet Manager to perform the following tasks:

- Create wallets
- Generate certificate requests
- Open wallets to access PKI-based services
- Save credentials to hardware security modules by using APIs which comply to Public-Key Cryptography Standard #11 specification (see PKCS #11)
- Upload wallets to and download them from an LDAP directory
- Import third-party PKCS #12-format wallets to use in an Oracle environment
- Export Oracle wallets to third-party environments

The following topics describe Oracle Wallet Manager features:

- Wallet Password Management
- Strong Wallet Encryption
- Microsoft Windows Registry Wallet Storage
- Backward Compatibility
- Third-Party Wallet Support
- LDAP Directory Support

11.1.1.1 Wallet Password Management

Oracle wallets are password protected. Oracle Wallet Manager includes an enhanced wallet password management module that enforces the following password management policy guidelines:

- Minimum password length (8 characters)
- Maximum password length unlimited
- Alphanumeric character mix required

11.1.1.2 Strong Wallet Encryption

Oracle Wallet Manager stores private keys associated with X.509 certificates and uses Triple-DES encryption.

11.1.1.3 Microsoft Windows Registry Wallet Storage

As an option, Oracle Wallet Manager enables you to store multiple Oracle wallets in the user profile area of the Microsoft Windows system registry or in a Windows file management system. Storing your wallets in the registry provides the following benefits:

- **Better Access Control.** Wallets stored in the user profile area of the registry are only accessible by the associated user. User access controls for the system thus become, by extension, access controls for the wallets. In addition, when a user logs out of a system, access to that user's wallets is effectively precluded.

- **Easier Administration.** Since wallets are associated with specific user profiles, no file permissions need to be managed, and the wallets stored in the profile are automatically deleted when the user profile is deleted. Oracle Wallet Manager can be used to create and manage the wallets in the registry.

11.1.1.3.1 Options Supported:

- Open wallet from the registry

- Save wallet to the registry

- Save As to a different registry location

- Delete wallet from the registry

- Open wallet from the file system and save it to the registry

- Open wallet from the registry and save it to the file system

11.1.1.4 Backward Compatibility

Oracle Wallet Manager is backward-compatible to Release 8.1.7 of the database.

11.1.1.5 Third-Party Wallet Support

Oracle Wallet Manager can use PKI credentials from the following third-party applications:

- Microsoft Internet Explorer 5.0 and later

- Netscape Communicator 4.7.2 and later

- OpenSSL

Browser PKI credential stores (those from Microsoft Internet Explorer and Netscape) hold user certificates, which contain the subject's public key and identifying information, and their associated trusted certificates. To use these credentials, you must export them from the third-party environment and save them in PKCS #12 format. Then you can use Oracle Wallet Manager to open them for use with SSL.

> **See Also:** "Section 11.1.5.1.3, "Importing Certificates Created with a Third-Party Tool"

11.1.1.6 LDAP Directory Support

Oracle Wallet Manager can upload wallets to and retrieve them from an LDAP-compliant directory. Storing wallets in a centralized LDAP-compliant directory lets users access them from multiple locations or devices, ensuring consistent and reliable user authentication while providing centralized wallet management throughout the wallet life cycle. To prevent accidental over-write of functional wallets, only wallets containing an installed certificate can be uploaded.

Directory user entries must be defined and configured in the LDAP directory before Oracle Wallet Manager can be used to upload or download wallets for a user. If a directory contains Oracle8*i* (or prior) users, they are automatically upgraded to use the wallet upload and download feature on first use.

Oracle Wallet Manager downloads a user wallet by using a simple password-based connection to the LDAP directory. However, for uploads it uses an SSL connection if the open wallet contains a certificate with SSL Oracle PKI certificate usage. If an SSL certificate is not present in the wallet, password-based authentication is used.

> **Note:** The directory password and the wallet password are independent, and can be different. Oracle recommends that these passwords be maintained to be consistently different, where neither one can logically be derived from the other.

See Also:

- Section 11.1.4.7, "Uploading a Wallet to an LDAP Directory"
- Section 11.1.4.8, "Downloading a Wallet from an LDAP Directory"
- Section 11.3.2, "Multiple Certificate Support"

11.1.2 Starting Oracle Wallet Manager

To start Oracle Wallet Manager:

- On Windows: Select **Start > Programs > Oracle -** *Oracle_Home_Name* **> Integrated Management Tools > Wallet Manager**

- On UNIX: At the command line, enter `owm`.

11.1.3 How to Create a Complete Wallet: Process Overview

A wallet is a necessary repository in which to securely store user certificates and the trust point**s** needed to validate the certificates of peers.

The following steps provide an overview of the complete wallet creation process:

1. Use Oracle Wallet Manager to create a new wallet:

   - See Section 11.1.4.1 for information about creating a wallet password.

   - See Section 11.1.4.2 for information about creating standard wallets (store credentials on your file system) and hardware security module wallets.

2. Generate a certificate request. Note that when you create a new wallet with Oracle Wallet Manager, the tool automatically prompts you to create a certificate request. See Section 11.1.5.1.1 for information about creating a certificate request.

3. Send the certificate request to the CA you want to use. You can copy and paste the certificate request text into an e-mail message, or you can export the certificate request to a file. See Section 11.1.5.1.7. Note that the certificate request becomes part of the wallet and must remain there until you remove its associated certificate.

4. When the CA sends your signed user certificate and its associated trusted certificate, then you can import these certificates in the following order. (Note that user certificates and trusted certificates in the PKCS #7 format can be imported at the same time.)

   - First import the CA's trusted certificate into the wallet. See Section 11.1.5.2.1. Note that this step may be optional if the new user certificate has been issued by one of the CAs whose trusted certificate is already present in Oracle Wallet Manager by default.

   - After you have successfully imported the trusted certificate, then import the user certificate that the CA sent to you into your wallet. See Section 11.1.5.1.2.

> **Note:** The BASE64 encoded PKCS#7 format used by most certificate authorities typically uses the following header and footer lines:
>
> ```
> -----BEGIN PKCS7-----
> -----END PKCS7-----
> ```
>
> Regular certificates contain the following header & footer lines:
>
> ```
> -----BEGIN CERTIFICATE-----
> -----END CERTIFICATE-----
> ```
>
> However, some certificate authorities use BEGIN CERTIFICATE and END CERTIFICATE header and footer lines in PKCS #7 format certificates as well. When certificates of PKCS #7 format are imported, the certificate authority certificates are imported as trusted certificates.
>
> If you import the user certificate without its certificate authority certificate, Oracle Wallet Manager prompts you for the certificate authority certificate that issued the user certificate.

5. (Optional) Set the auto login feature for the wallet. See Section 11.1.4.14.

   Typically, this feature, which enables PKI-based access to services without a password, is required for most wallets. It is required for database server and client wallets. It is only optional for products that take the wallet password at the time of startup.

After completing the preceding process, you have a wallet that contains a user certificate and its associated trust points.

11.1.4 Managing Wallets

This section describes how to create a new wallet and perform associated wallet management tasks in the following topics:

- Required Guidelines for Creating Wallet Passwords
- Creating a New Wallet
- Opening an Existing Wallet
- Closing a Wallet
- Exporting Oracle Wallets to Third-Party Environments
- Exporting Oracle Wallets to Tools That Do Not Support PKCS #12
- Uploading a Wallet to an LDAP Directory
- Downloading a Wallet from an LDAP Directory
- Saving Changes
- Saving the Open Wallet to a New Location
- Saving in System Default
- Deleting the Wallet
- Changing the Password
- Using Auto Login

11.1.4.1 Required Guidelines for Creating Wallet Passwords

Because an Oracle wallet contains user credentials that can be used to authenticate the user to multiple databases, it is especially important to choose a strong wallet password. A malicious user who guesses the wallet password can access all the databases to which the wallet owner has access.

Passwords must contain at least eight characters that consist of alphabetic characters combined with numbers or special characters.

> **Caution:** It is strongly recommended that users avoid choosing easily guessed passwords based on user names, phone numbers, or government identification numbers, such as "admin0," "oracle1," or "2135551212A." This prevents a potential attacker from using personal information to deduce the users' passwords. It is also a prudent security practice for users to change their passwords periodically, such as once in each month or once in each quarter.
>
> When you change passwords, you must regenerate auto login wallets.

> **See Also:**
> - Section 11.1.1.1, "Wallet Password Management"
> - Section 11.1.4.14, "Using Auto Login"

11.1.4.2 Creating a New Wallet

You can use Oracle Wallet Manager to create PKCS #12 wallets (the standard default wallet type) that store credentials in a directory on your file system. It can also be used to create PKCS #11 wallets that store credentials on a hardware security module for servers, or private keys on tokens for clients. The following sections explain how to create both types of wallets by using Oracle Wallet Manager.

11.1.4.2.1 Creating a Standard Wallet Unless you have a hardware security module (a PKCS #11 device), then you should use a standard wallet that stores credentials in a directory on your file system.

To create a standard wallet, perform the following tasks:

1. Choose **Wallet** > **New** from the menu bar. The New Wallet dialog box appears.

2. Follow the Section 11.1.4.1 and enter a password in the **Wallet Password** field. This password protects unauthorized use of your credentials.

3. Re-enter that password in the **Confirm Password** field.

4. Choose **Standard** from the **Wallet Type** list.

5. Click **OK** to continue. If the entered password does not conform to the required guidelines, then the following message appears:

```
Password must have a minimum length of eight characters,
and contain alphabetic characters combined with numbers
or special characters.
Do you want to try again?
```

6. An alert is displayed that informs you that a new empty wallet has been created. It prompts you to decide whether you want to add a certificate request. See Section 11.1.5.1.1.

If you choose **No**, you are returned to the Oracle Wallet Manager main window. The new wallet you just created appears in the left window pane. The certificate has a status of **[Empty]**, and the wallet displays its default trusted certificates.

7. Select **Wallet** > **Save In System Default** to save the new wallet.

If you do not have permission to save the wallet in the system default, you can save it to another location. This location must be used in the SSL configuration for clients and servers.

A message at the bottom of the window confirms that the wallet was successfully saved.

11.1.4.2.2 Creating a Wallet to Store Hardware Security Module Credentials To create a wallet to store credentials on a hardware security module that complies with PKCS #11, perform the following tasks:

1. Choose **Wallet** > **New** from the menu bar; the New Wallet dialog box appears.

2. Follow Section 11.1.4.1 and enter a password in the **Wallet Password** field.

3. Re-enter that password in the **Confirm Password** field.

4. Choose **PKCS11** from the **Wallet Type** list, and click **OK** to continue. The New PKCS11 Wallet window appears.

5. Choose a vendor name from the **Select Hardware Vendor** list.

> **Note:** In the current release of Oracle Wallet Manager, only nCipher hardware has been certified to interoperate with Oracle wallets.

6. In the **PKCS11 library filename** field, enter the path to the directory in which the PKCS11 library is stored, or click **Browse** to find it by searching the file system.

7. Enter the **SmartCard password**, and choose **OK**.

The smart card password, which is different from the wallet password, is stored in the wallet.

8. An alert is displayed, and informs you that a new empty wallet has been created. It prompts you to decide whether you want to add a certificate request. See Section 11.1.5.1.1.

If you choose **No**, you are returned to the Oracle Wallet Manager main window. The new wallet you just created appears in the left window pane. The certificate has a status of **[Empty]**, and the wallet displays its default trusted certificates.

9. Select **Wallet** > **Save In System Default** to save the new wallet.

If you do not have permission to save the wallet in the system default, you can save it to another location.

A message at the bottom of the window confirms that the wallet was successfully saved.

> **Note:** If you change the SmartCard password or move the PKCS #11 library, an error message displays when you try to open the wallet. Then you are prompted to enter the new SmartCard password or the new path to the library.

11.1.4.3 Opening an Existing Wallet

Open a wallet that already exists in the file system directory as follows:

1. Choose **Wallet > Open** from the menu bar. The Select Directory dialog box appears.

2. Navigate to the directory location in which the wallet is located, and select the directory.

3. Choose **OK**. The Open Wallet dialog box appears.

4. Enter the wallet password in the **Wallet Password** field.

5. Choose **OK**.

    You are returned to the main window and a message appears at the bottom of the window indicating the wallet was opened successfully. The wallet's certificate and its trusted certificates are displayed in the left window pane.

11.1.4.4 Closing a Wallet

To close an open wallet in the currently selected directory:

Choose **Wallet > Close**.

A message appears at the bottom of the window to confirm that the wallet is closed.

11.1.4.5 Exporting Oracle Wallets to Third-Party Environments

Oracle Wallet Manager can export its own wallets to third party environments.

To export a wallet to third-party environments:

1. Use Oracle Wallet Manager to save the wallet file.

2. Follow the procedure specific to your third-party product to import an operating system PKCS #12 wallet file created by Oracle Wallet Manager (called `ewallet.p12` on UNIX and Windows platforms).

    > **Note:**
    >
    > - Oracle Wallet Manager supports multiple certificates for each wallet, yet current browsers typically support import of single-certificate wallets only. For these browsers, you must export an Oracle wallet containing a single key-pair.
    >
    > - Oracle Wallet Manager supports wallet export to only Netscape Communicator 4.7.2 and later, OpenSSL, and Microsoft Internet Explorer 5.0 and later.

11.1.4.6 Exporting Oracle Wallets to Tools That Do Not Support PKCS #12

You can export a wallet to a text-based PKI format if you want to put a wallet into a tool that does not support PKCS #12. Individual components are formatted according to the standards listed in Table 11–1. Within the wallet, only those certificates with SSL key usage are exported with the wallet.

To export a wallet to text-based PKI format:

1. Choose **Operations > Export Wallet**. The Export Wallet dialog box appears.

2. Enter the destination file system directory for the wallet, or navigate to the directory structure under **Folders**.

3. Enter the destination file name for the wallet.

4. Choose **OK** to return to the main window.

Table 11–1   PKI Wallet Encoding Standards

Component	Encoding Standard
Certificate chains	X509v3
Trusted certificates	X509v3
Private keys	PKCS #8

11.1.4.7 Uploading a Wallet to an LDAP Directory

To upload a wallet to an LDAP directory, Oracle Wallet Manager uses SSL if the specified wallet contains an SSL certificate. Otherwise, it lets you enter the directory password.

To prevent accidental destruction of your wallet, Oracle Wallet Manager will not permit you to execute the upload option unless the target wallet is currently open and contains at least one user certificate.

To upload a wallet:

1. Choose **Wallet > Upload Into The Directory Service**. If the currently open wallet has not been saved, a dialog box appears with the following message:

Wallet needs to be saved before uploading.

Choose **Yes** to proceed.

2. Wallet certificates are checked for SSL key usage. Depending on whether a certificate with SSL key usage is found in the wallet, one of the following results occur:

- **If at least one certificate has SSL key usage:** When prompted, enter the LDAP directory server hostname and port information, then click **OK**. Oracle Wallet Manager attempts to connect to the LDAP directory server using SSL. A message appears indicating whether the wallet was uploaded successfully or it failed.

- **If no certificates have SSL key usage:** When prompted, enter the user's distinguished name (DN), the LDAP server hostname and port information, and click **OK**. Oracle Wallet Manager attempts to connect to the LDAP directory server using simple password authentication mode, assuming that the wallet password is the same as the directory password.

  If the connection fails, a dialog box prompts for the directory password of the specified DN. Oracle Wallet Manager attempts to connect to the LDAP directory server using this password and displays a warning message if the attempt fails. Otherwise, Oracle Wallet Manager displays a status message at the bottom of the window indicating that the upload was successful.

11.1.4.8 Downloading a Wallet from an LDAP Directory

When a wallet is downloaded from an LDAP directory, it is resident in working memory. It is not saved to the file system unless you expressly save it using any of the Save options described in the following sections.

See Also:

-
-
-

To download a wallet from an LDAP directory:

1. Choose **Wallet > Download From The Directory Service**.

2. A dialog box prompts for the user's distinguished name (DN), and the LDAP directory password, hostname, and port information. Oracle Wallet Manager uses simple password authentication to connect to the LDAP directory.

   Depending on whether the downloading operation succeeds or not, one of the following results occurs:

   - **If the download operation fails:** Check to make sure that you have correctly entered the user's DN, and the LDAP server hostname and port information.

   - **If the download is successful:** Choose **OK** to open the downloaded wallet. Oracle Wallet Manager attempts to open that wallet using the directory password. If the operation fails after using the directory password, then a dialog box prompts for the wallet password.

     If Oracle Wallet Manager cannot open the target wallet using the wallet password, then check to make sure you entered the correct password. Otherwise a message displays at the bottom of the window, indicating that the wallet was downloaded successfully.

11.1.4.9 Saving Changes

To save your changes to the current open wallet:

Choose **Wallet** > **Save**.

A message at the bottom of the window confirms that the wallet changes were successfully saved to the wallet in the selected directory location.

11.1.4.10 Saving the Open Wallet to a New Location

To save open wallets to a new location, use the **Save As** menu option:

1. Choose **Wallet** > **Save As**. The Select Directory dialog box appears.

2. Select a directory location in which to save the wallet.

3. Choose **OK**.

   The following message appears if a wallet already exists in the selected location:

   ```
   A wallet already exists in the selected path. Do you want to overwrite it?
   ```

   Choose **Yes** to overwrite the existing wallet, or **No** to save the wallet to another location.

   A message at the bottom of the window confirms that the wallet was successfully saved to the selected directory location.

11.1.4.11 Saving in System Default

To save wallets in the default directory location, use the **Save In System Default** menu option:

Choose **Wallet** > **Save In System Default**.

A message at the bottom of the window confirms that the wallet was successfully saved in the system default wallet location as follows for UNIX and Windows platforms:

- On UNIX: `/etc/ORACLE/WALLETS/$USER/`

- On Windows: `%USERPROFILE%\ORACLE\WALLETS\`

> **Note:**
>
> - SSL uses the wallet that is saved in the system default directory location.
>
> - Some Oracle applications are not able to use the wallet if it is not in the system default location. Check the Oracle documentation for your specific application to determine whether wallets must be placed in the default wallet directory location.

11.1.4.12 Deleting the Wallet

To delete the current open wallet:

1. Choose **Wallet** > **Delete**. The Delete Wallet dialog box appears.

2. Review the displayed wallet location to verify you are deleting the correct wallet.

3. Enter the wallet password.

4. Choose **OK**. A dialog panel appears to inform you that the wallet was successfully deleted.

> **Note:** Any open wallet in application memory will remain in memory until the application exits. Therefore, deleting a wallet that is currently in use does not immediately affect system operation.

11.1.4.13 Changing the Password

A password change is effective immediately. The wallet is saved to the currently selected directory, with the new encrypted password.

> **Note:** If you are using a wallet with auto login enabled, you must regenerate the auto login wallet after changing the password. See Section 11.1.4.14, "Using Auto Login"

To change the password for the current open wallet:

1. Choose **Wallet** > **Change Password**. The Change Wallet Password dialog box appears.

2. Enter the existing wallet password.

3. Enter the new password.

4. Re-enter the new password.

5. Choose **OK**.

A message at the bottom of the window confirms that the password was successfully changed.

> **See Also:**
>
> - Section 11.1.4.1, "Required Guidelines for Creating Wallet Passwords"
> - Section 11.1.1.1, "Wallet Password Management"

11.1.4.14 Using Auto Login

The Oracle Wallet Manager auto login feature creates an obfuscated copy of the wallet and enables PKI-based access to services without a password until the auto login feature is disabled for the wallet. File system permissions provide the necessary security for auto-login wallets.

You must enable auto login if you want single sign-on access to multiple Oracle databases, which is disabled by default. Sometimes these are called "SSO wallets" because they provide single sign-on capability.

11.1.4.14.1   Enabling Auto Login To enable auto login:

1. Choose **Wallet** from the menu bar.

2. Check **Auto Login**. A message at the bottom of the window indicates that auto login is enabled.

11.1.4.14.2   Disabling Auto Login To disable auto login:

1. Choose **Wallet** from the menu bar.

2. Uncheck **Auto Login**. A message at the bottom of the window indicates that auto login is disabled.

11.1.5 Managing Certificates

Oracle Wallet Manager uses two kinds of certificates: user certificates and trusted certificates. All certificates are signed data structures that bind a network identity with a corresponding public key. User certificates are used by end entities, including server applications, to validate an end entity's identity in a public key/private key exchange. In comparison, trusted certificates are any certificates that you trust, such as those provided by CAs to validate the user certificates that they issue.

This section describes how to manage both certificate types, in the following subtopics:

- Managing User Certificates
- Managing Trusted Certificates

> **Note:**   Before a user certificate can be installed, the wallet must contain the trusted certificate representing the certificate authority who issued that user certificate. However, whenever you create a new wallet, several publicly trusted certificates are automatically installed, since they are so widely used. If the necessary certificate authority is not represented, you must install its certificate first.
>
> Also, you can import using the PKCS#7 certificate chain format, which gives you the user certificate and the CA certificate at the same time.

11.1.5.1 Managing User Certificates

User certificates can be used by end users, smart cards, or applications, such as Web servers. Server certificates are a type of user certificate. For example, if a CA issues a certificate for a Web server, placing its distinguished name (DN) in the Subject field, then the Web server is the certificate owner, thus the "user" for this user certificate.

Managing user certificates involves the following tasks:

- Adding a Certificate Request
- Importing the User Certificate into the Wallet
- Importing Certificates Created with a Third-Party Tool
- Removing a User Certificate from a Wallet
- Removing a Certificate Request
- Exporting a User Certificate
- Exporting a User Certificate Request

11.1.5.1.1 Adding a Certificate Request You can add multiple certificate requests with Oracle Wallet Manager. When adding multiple requests, Oracle Wallet Manager automatically populates each subsequent request dialog box with the content of the initial request that you can then edit.

The actual certificate request becomes part of the wallet. You can reuse any certificate request to obtain a new certificate. However, you cannot edit an existing certificate request. Store only a correctly filled out certificate request in a wallet.

To create a PKCS #10 certificate request:

1. Choose **Operations** > **Add Certificate Request**. The Add Certificate Request dialog box appears.

2. Enter the information specified in Table 11–2.

3. Choose **OK**. A message informs you that a certificate request was successfully created. You can either copy the certificate request text from the body of this dialog panel and paste it into an e-mail message to send to a certificate authority, or you can export the certificate request to a file.

4. Choose **OK** to return to the Oracle Wallet Manager main window. The status of the certificate changes to **[Requested]**.

> **See Also:** "Section 11.1.5.1.7, "Exporting a User Certificate Request"

Table 11–2   Certificate Request: Fields and Descriptions

Field Name	Description
Common Name	Mandatory. Enter the name of the user's or service's identity. Enter a user's name in first name /last name format. Example: Eileen.Sanger
Organizational Unit	Optional. Enter the name of the identity's organizational unit. Example: Finance.
Organization	Optional. Enter the name of the identity's organization. Example: XYZ Corp.
Locality/City	Optional. Enter the name of the locality or city in which the identity resides.

Table 11–2   (Cont.) Certificate Request: Fields and Descriptions

Field Name	Description
State/Province	Optional. Enter the full name of the state or province in which the identity resides.
	Enter the full state name, because some certificate authorities do not accept two–letter abbreviations.
Country	Mandatory. Choose to view a list of country abbreviations. Select the country in which the organization is located.
Key Size	Mandatory. Choose to view a list of key sizes to use when creating the public/private key pair. See Table 11–3 to evaluate key size.
Advanced	Optional. Choose **Advanced** to view the Advanced Certificate Request dialog panel. Use this field to edit or customize the identity's distinguished name (DN). For example, you can edit the full state name and locality.

Table 11–3 lists the available key sizes and the relative security each size provides. Typically, CAs use key sizes of 1024 or 2048. When certificate owners want to keep their keys for a longer duration, they choose 3072 or 4096 bit keys.

Table 11–3   Available Key Sizes

Key Size	Relative Security Level
512 or 768	Not regarded as secure.
1024 or 2048	Secure.
3072 or 4096	Very secure.

11.1.5.1.2   Importing the User Certificate into the Wallet When the Certificate Authority grants you a certificate, it may send you an e-mail that contains your certificate in text (BASE64) form or attached as a binary file.

> **Note:**   Certificate authorities may send your certificate in a PKCS #7 certificate chain or as an individual X.509 certificate. Oracle Wallet Manager can import both types.
>
> PKCS #7 certificate chains are a collection of certificates, including the user's certificate and all of the supporting trusted CA and subCA certificates.
>
> In contrast, an X.509 certificate file contains an individual certificate without the supporting certificate chain.
>
> However, before you can import any such individual certificate, the signer's certificate must be a Trusted Certificate in the wallet.

To import the user certificate from the text of the Certificate Authority's e-mail, copy the certificate, represented as text (BASE64), from the certificate authority's e-mail message. Include the lines `Begin Certificate` and `End Certificate`.

1. Choose **Operations** > **Import User Certificate**. The Import Certificate dialog box appears.

2. Choose **Paste the certificate**, and then click **OK**. Another Import Certificate dialog box appears with the following message:

```
Please provide a base64 format certificate and paste it below.
```

3. Paste the certificate into the dialog box, and choose **OK**.

   a. If the certificate received is in PKCS#7 format, it is installed, and all the other certificates included with the PKCS#7 data are placed in the Trusted Certificate list.

   b. If the certificate received is *not* in PKCS#7 format, and the certificate of its CA is not already in the Trusted Certificates list, then more must be done. Oracle Wallet Manager will ask you to import the certificate of the CA that issued your certificate. This CA certificate will be placed in the Trusted Certificates list. (If the CA certificate was already in the Trusted Certificates list, your certificate is imported without additional steps.)

   After either (a) or (b) succeeds, a message at the bottom of the window confirms that the certificate was successfully installed. The Oracle Wallet Manager main window reappears, and the status of the corresponding entry in the left panel subtree changes to **[Ready]**.

   Note:

   The standard X.509 certificate includes the following start and end text:

   ```
   -----BEGIN CERTIFICATE-----
   -----END CERTIFICATE-----
   ```

   A typical PKCS#7 certificate includes more, as described earlier, and includes the following start and end text:

   ```
   -----BEGIN PKCS7-----
   -----END PKCS7-----
   ```

   You can use the standard Ctrl+c to copy, including all dashes, and Ctrl+v to paste.

To import the certificate from a file:

The user certificate in the file can be in either text (BASE64) or binary (`der`) format.

1. Choose **Operations** > **Import User Certificate**. The Import Certificate dialog box appears.

2. Choose **Select a file that contains the certificate**, and click **OK**. Another Import Certificate dialog box appears.

3. Enter the path or folder name of the certificate file location.

4. Select the name of the certificate file (for example, `cert.txt`, `cert.der`).

5. Choose **OK**.

   a. If the certificate received is in PKCS#7 format, it is installed, and all the other certificates included with the PKCS#7 data are placed in the Trusted Certificate list.

   b. If the certificate received is *not* in PKCS#7 format, and the certificate of its CA is not already in the Trusted Certificates list, then more must be done. Oracle Wallet Manager will ask you to import the certificate of the CA that issued your certificate. This CA certificate will be placed in the Trusted Certificates list. (If the CA certificate was already in the Trusted Certificates list, your certificate is imported without additional steps.)

After either (a) or (b) succeeds, a message at the bottom of the window confirms that the certificate was successfully installed. You are returned to the Oracle Wallet Manager main panel, and the status of the corresponding entry in the left panel subtree changes to **[Ready.]**

11.1.5.1.3 Importing Certificates Created with a Third-Party Tool Third-party certificates are those created from certificate requests that were not generated using Oracle Wallet Manager. These third-party certificates are actually wallets, in the Oracle sense, because they contain more than just the user certificate; they also contain the private key for that certificate. Furthermore, they include the chain of trusted certificates validating that the certificate was created by a trustworthy entity.

Oracle Wallet Manager makes these wallets available in a single step by importing them in PKCS#12 format, which includes all three elements described earlier: the user certificate, the private key, and the trusted certificates. It supports the following PKCS #12-format certificates:

- Netscape Communicator 4.x

- Microsoft Internet Explorer 5.x and later

Oracle Wallet Manager adheres to the PKCS#12 standard, so certificates exported by any PKCS#12-compliant tool should be usable with Oracle Wallet Manager.

Such third-party certificates cannot be stored into existing Oracle wallets because they would lack the private key and chain of trusted authorities. Therefore, each such certificate is exported and retrieved instead as an independent PKCS#12 file, that is, as its own wallet.

To import a certificate created with a third-party tool, you must first export it from the application you are using, and then save it as a wallet file that can be read by Oracle Wallet Manager.

To import a certificate created with a third-party tool, perform the following steps:

1. Follow the procedures for your particular product to export the certificate. Take the actions indicated in the exporting product to include the private key in the export, and specify the new password to protect the exported certificate. Also include all associated trust points. (Under PKCS #12, browsers do not necessarily export **trusted certificate**s, other than the signer's own certificate. You may need to add additional certificates to authenticate to your peers. You can use Oracle Wallet Manager to import trusted certificates.)

    The resulting file, containing the certificate, the private key, and the trust points, is the new wallet that enables the third-party certificate to be used.

2. Save the exported certificate to a file name appropriate for your operating system in a directory expected by Oracle Wallet Manager.

    For UNIX and Windows, the appropriate file name is `ewallet.p12`.

    For other operating systems, see the Oracle documentation for the applicable operating system.

3. Use Oracle Wallet Manager to navigate to the directory in which you saved the `ewallet.p12` file and open it to use the PKI credentials it contains.

> **Note:** The password will be required whenever the associated
> application starts up or otherwise needs the certificate. To make such
> access automatic, see Section 11.1.4.14, "Using Auto Login".
>
> However, if the private key for the desired certificate is held in a
> separate hardware security module, you will not be able to import
> that certificate.

If you exported the trusted certificate separately, then you must import the trusted
certificate first before you open the `ewallet.p12` file that contains the imported
third-party user certificate.

> **See Also:** "Section 11.1.5.2.1, "Importing a Trusted Certificate"

11.1.5.1.4 Removing a User Certificate from a Wallet To remove a user certificate from a
wallet:

1. In the left panel subtree, select the certificate that you want to remove.

2. Choose **Operations** > **Remove User Certificate**. A dialog panel appears and
   prompts you to verify that you want to remove the user certificate from the wallet.

3. Choose **Yes** to return to the Oracle Wallet Manager main panel. The certificate
   displays a status of **[Requested].**

11.1.5.1.5 Removing a Certificate Request You must remove a certificate before removing
its associated request.

To remove a certificate request:

1. In the left panel subtree, select the certificate request that you want to remove.

2. Choose **Operations** > **Remove Certificate Request**.

3. Click **Yes**. The certificate displays a status of **[Empty].**

11.1.5.1.6 Exporting a User Certificate To save the certificate in a file system directory,
export the certificate by using the following steps:

1. In the left panel subtree, select the certificate that you want to export.

2. Choose **Operations** > **Export User Certificate** from the menu bar. The Export
   Certificate dialog box appears.

3. Enter the file system directory location in which you want to save your certificate,
   or navigate to the directory structure under **Folders**.

4. Enter a file name for your certificate in the **Enter File Name** field.

5. Choose **OK**. A message at the bottom of the window confirms that the certificate
   was successfully exported to the file. You are returned to the Oracle Wallet
   Manager main window.

> **See Also:** "Section 11.1.4.5, "Exporting Oracle Wallets to Third-Party
> Environments" for information about exporting wallets. Note that
> Oracle Wallet Manager supports storing multiple certificates in a
> single wallet, yet current browsers typically support only
> single-certificate wallets. For these browsers, you must export an
> Oracle wallet that contains a single key-pair.

11.1.5.1.7 Exporting a User Certificate Request To save the certificate request in a file system directory, export the certificate request by using the following steps:

1. In the left panel subtree, select the certificate request that you want to export.

2. Choose **Operations** > **Export Certificate Request**. The Export Certificate Request dialog box appears.

3. Enter the file system directory location in which you want to save your certificate request, or navigate to the directory structure under **Folders**.

4. Enter a file name for your certificate request, in the **Enter File Name** field.

5. Choose **OK**. A message at the bottom of the window confirms that the certificate request was successfully exported to the file. You are returned to the Oracle Wallet Manager main window.

11.1.5.2 Managing Trusted Certificates

Managing trusted certificates includes the following tasks:

- Importing a Trusted Certificate

- Removing a Trusted Certificate

- Exporting a Trusted Certificate

- Exporting All Trusted Certificates

11.1.5.2.1 Importing a Trusted Certificate You can import a trusted certificate into a wallet in either of two ways: paste the trusted certificate from an e-mail that you receive from the certificate authority, or import the trusted certificate from a file.

Oracle Wallet Manager automatically installs trusted certificates from VeriSign, RSA, Entrust, and GTE CyberTrust when you create a new wallet.

To copy and paste the text only (BASE64) trusted certificate:

Copy the trusted certificate from the body of the e-mail message you received that contained the user certificate. Include the lines `Begin Certificate` and `End Certificate`.

1. Choose **Operations** > **Import Trusted Certificate** from the menu bar. The Import Trusted Certificate dialog panel appears.

2. Choose **Paste the Certificate**, and click **OK**. Another Import Trusted Certificate dialog panel appears with the following message:

   ```
   Please provide a base64 format certificate and paste it below.
   ```

3. Paste the certificate into the window, and click **OK**. A message at the bottom of the window informs you that the trusted certificate was successfully installed.

4. Choose **OK**. You are returned to the Oracle Wallet Manager main panel, and the trusted certificate appears at the bottom of the Trusted Certificates tree.

   Keyboard shortcuts for copying and pasting certificates:

   Use Ctrl+c to copy, and use Ctrl+v to paste.

To import a file that contains the trusted certificate:

The file containing the trusted certificate should have been saved in either text (BASE64) or binary (`der`) format.

1. Choose **Operations** > **Import Trusted Certificate**. The Import Trusted Certificate dialog panel appears.

2. Enter the path or folder name of the trusted certificate location.

3. Select the name of the trusted certificate file (for example, `cert.txt`).

4. Choose **OK**. A message at the bottom of the window informs you that the trusted certificate was successfully imported into the wallet.

5. Choose **OK** to exit the dialog panel. You are returned to the Oracle Wallet Manager main panel, and the trusted certificate appears at the bottom of the Trusted Certificates tree.

11.1.5.2.2 Removing a Trusted Certificate You cannot remove a trusted certificate if it has been used to sign a user certificate still present in the wallet. To remove such trusted certificates, you must first remove the certificates it has signed. Also, you cannot verify a certificate after its trusted certificate has been removed from your wallet.

To remove a trusted certificate from a wallet:

1. Select the trusted certificate listed in the Trusted Certificates tree.

2. Choose **Operations** > **Remove Trusted Certificate** from the menu bar.

   A dialog panel warns you that your user certificate will no longer be verifiable by its recipients if you remove the trusted certificate that was used to sign it.

3. Choose **Yes**. The selected trusted certificate is removed from the Trusted Certificates tree.

11.1.5.2.3 Exporting a Trusted Certificate To export a trusted certificate to another file system location:

1. In the left panel subtree, select the trusted certificate that you want to export.

2. Select **Operations** > **Export Trusted Certificate**. The Export Trusted Certificate dialog box appears.

3. Enter a file system directory in which you want to save your trusted certificate, or navigate to the directory structure under **Folders**.

4. Enter a file name to save your trusted certificate.

5. Choose **OK**. You are returned to the Oracle Wallet Manager main window.

11.1.5.2.4 Exporting All Trusted Certificates To export all of your trusted certificates to another file system location:

1. Choose **Operations** > **Export All Trusted Certificates**. The Export Trusted Certificate dialog box appears.

2. Enter a file system directory location in which you want to save your trusted certificates, or navigate to the directory structure under **Folders**.

3. Enter a file name to save your trusted certificates.

4. Choose **OK**. You are returned to the Oracle Wallet Manager main window.

11.2 Performing Certificate Validation and CRL Management with the orapki Utility

The `orapki` utility is a command-line tool that you can use to manage certificate revocation lists (CRLs), to create and manage Oracle wallets, and to create signed certificates for testing purposes.

The following topics describe this tool and how to use it:

- orapki Overview
- Displaying orapki Help
- Creating Signed Certificates for Testing Purposes
- Managing Oracle Wallets with the orapki Utility
- Managing Certificate Revocation Lists (CRLs) with orapki Utility
- orapki Utility Commands Summary

11.2.1 orapki Overview

The `orapki` utility is provided to manage public key infrastructure (PKI) elements, such as wallets and certificate revocation lists, on the command line so the tasks it performs can be incorporated into scripts. This enables you to automate many of the routine tasks of maintaining a PKI.

This command-line utility can be used to perform the following tasks:

- Creating signed certificates for testing purposes
- Managing Oracle wallets:
  - Creating and displaying Oracle wallets
  - Adding and removing certificate requests
  - Adding and removing certificates
  - Adding and removing trusted certificates
- Managing certificate revocation lists (CRLs):
  - Renaming CRLs with a hash value for certificate validation
  - Uploading, listing, viewing, and deleting CRLs in Oracle Internet Directory

11.2.1.1 orapki Utility 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
```

11.2.2 Displaying orapki Help

You can display all the `orapki` commands that are available for a specific mode by entering the following at the command line:

```
orapki mode help
```

For example, to display all available commands for managing certificate revocation lists (CRLs), enter the following at the command line:

```
orapki CRL help
```

> **Note:** Using the `-summary`, `-complete`, or `-wallet` command options is always optional. A command will still run if these command options are not specified.

11.2.3 Creating Signed Certificates for Testing Purposes

This command-line utility provides a convenient, lightweight way to create signed certificates for testing purposes. The following syntax can be used to create signed certificates and to view certificates:

To create a signed certificate for testing purposes:

```
orapki cert create [-wallet wallet_location] -request
 certificate_request_location
-cert certificate_location -validity number_of_days [-summary]
```

This command creates a signed certificate from the certificate request. The `-wallet` parameter specifies the wallet containing the user certificate and private key that will be used to sign the certificate request. The `-validity` parameter specifies the number of days, starting from the current date, that this certificate will be valid. Specifying a certificate and certificate request is mandatory for this command.

To view a certificate:

```
orapki cert display -cert certificate_location [-summary | -complete]
```

This command enables you to view a test certificate that you have created with `orapki`. You can choose either `-summary` or `-complete`, which determines how much detail the command will display. If you choose `-summary`, the command will display the certificate and its expiration date. If you choose `-complete`, it will display additional certificate information, including the serial number and public key.

11.2.4 Managing Oracle Wallets with the orapki Utility

The following sections describe the syntax used to create and manage Oracle wallets with the `orapki` command-line utility. You can use these `orapki` utility `wallet` module commands in scripts to automate the wallet creation process.

- Creating and Viewing Oracle Wallets with orapki
- Adding Certificates and Certificate Requests to Oracle Wallets with orapki
- Exporting Certificates and Certificate Requests from Oracle Wallets with orapki

> **Note:** The `-wallet` parameter is mandatory for all `wallet` module commands.

11.2.4.1 Creating and Viewing Oracle Wallets with orapki

To create an Oracle wallet:

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

This command will prompt you to enter and re-enter a wallet password. It creates a wallet in the location specified for -wallet.

To create an Oracle wallet with auto login enabled:

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

This command creates a wallet with auto login enabled, or it can also be used to enable auto login on an existing wallet. If the wallet_location already contains a wallet, then auto login will be enabled for it. To disable the auto login feature, use Oracle Wallet Manager. See Section 11.1.4.14 for details.

> **Note:** For wallets with the auto login feature enabled, you are prompted for a password only for operations that modify the wallet, such as add.

To view an Oracle wallet:

```
orapki wallet display -wallet wallet_location
```

This command displays the certificate requests, user certificates, and trusted certificates contained in the wallet.

11.2.4.2 Adding Certificates and Certificate Requests to Oracle Wallets with orapki

To add a certificate request to an Oracle wallet:

```
orapki wallet add -wallet wallet_location -dn user_dn -keySize 512|1024|2048
```

This command adds a certificate request to a wallet for the user with the specified distinguished name (user_dn). The request also specifies the requested certificate's key size (512, 1024, or 2048 bits). To sign the request, export it with the export option. See Section 11.2.4.3.

To add a trusted certificate to an Oracle wallet:

```
orapki wallet add -wallet wallet_location -trusted_cert -cert
certificate_location
```

This command adds a trusted certificate, at the specified location (-cert certificate_location), to a wallet. You must add all trusted certificates in the certificate chain of a user certificate before adding a user certificate, or the command to add the user certificate will fail.

To add a root certificate to an Oracle wallet:

```
orapki wallet add -wallet wallet_location -dn
certificate_dn -keySize 512|1024|2048 -self_signed -validity number_of_days
```

This command creates a new self-signed (root) certificate and adds it to the wallet. The -validity parameter (mandatory) specifies the number of days, starting from the current date, that this certificate will be valid. You can specify a key size for this root certificate (-keySize) of 512, 1024, or 2048 bits.

To add a user certificate to an Oracle wallet:

```
orapki wallet add -wallet wallet_location -user_cert -cert certificate_location
```

This command adds the user certificate at the location specified with the `-cert` parameter to the Oracle wallet at the `wallet_location`. Before you add a user certificate to a wallet, you must add all the trusted certificates that make up the certificate chain. If all trusted certificates are not installed in the wallet before you add the user certificate, then adding the user certificate will fail.

11.2.4.3 Exporting Certificates and Certificate Requests from Oracle Wallets with orapki

To export a certificate from an Oracle wallet:

```
orapki wallet export -wallet wallet_location -dn
certificate_dn -cert certificate_filename
```

This command exports a certificate with the subject's distinguished name (`-dn`) from a wallet to a file that is specified by `-cert`.

To export a certificate request from an Oracle wallet:

```
orapki wallet export -wallet wallet_location -dn
certificate_request_dn -request certificate_request_filename
```

This command exports a certificate request with the subject's distinguished name (`-dn`) from a wallet to a file that is specified by `-request`.

11.2.5 Managing Certificate Revocation Lists (CRLs) with orapki Utility

CRLs must be managed with `orapki`. This utility creates a hashed value of the CRL issuer's name to identify the CRLs location in your system. If you do not use `orapki`, your Oracle server cannot locate CRLs to validate PKI digital certificates. The following sections describe CRLs, how you use them, and how to use orapki to manage them:

- Section 11.2.5.1, "About Certificate Validation with Certificate Revocation Lists"

- Section 11.2.5.2, "Certificate Revocation List Management"

11.2.5.1 About Certificate Validation with Certificate Revocation Lists

The process of determining whether a given certificate can be used in a given context is referred to as certificate validation. Certificate validation includes determining that:

- A trusted certificate authority (CA) has digitally signed the certificate.

- The certificate's digital signature corresponds to the independently-calculated hash value of the certificate itself and the certificate signer's (CA's) public key.

- The certificate has not expired.

- The certificate has not been revoked.

The SSL network layer automatically performs the first three validation checks, but you must configure certificate revocation list (CRL) checking to ensure that certificates have not been revoked. CRLs are signed data structures that contain a list of revoked certificates. They are usually issued and signed by the same entity who issued the original certificate.

11.2.5.1.1 What CRLs Should You Use? You should have CRLs for all of the trust points that you honor. The trust points are the trusted certificates from a third-party identity that is qualified with a level of trust. Typically, the certificate authorities you trust are called trust points.

11.2.5.1.2 How CRL Checking Works Certificate revocation status is checked against CRLs which are located in file system directories, Oracle Internet Directory, or downloaded from the location specified in the CRL Distribution Point (CRL DP) extension on the certificate. If you store your CRLs on the local file system or in the directory, then you must update them regularly. If you use CRL DPs then CRLs are downloaded each time a certificate is used so there is no need to regularly refresh the CRLs.

The server searches for CRLs in the following locations in the order listed. When the system finds a CRL that matches the certificate CA's DN, it stops searching.

1. Local file system

    The system checks the `sqlnet.ora` file for the `SSL_CRL_FILE` parameter first, followed by the `SSL_CRL_PATH` parameter. If these two parameters are not specified, then the system checks the wallet location for any CRLs.

    Note: if you store CRLs on your local file system, then you must use the `orapki` utility to periodically update them. See Section 11.2.5.2.1.

2. Oracle Internet Directory

    If the server cannot locate the CRL on the local file system and directory connection information has been configured in the `ORACLE_HOME/ldap/admin/ldap.ora` file, then the server searches in the directory. It searches the CRL subtree by using the CA's distinguished name (DN) and the DN of the CRL subtree.

    The server must have a properly configured `ldap.ora` file to search for CRLs in the directory. It cannot use the Domain Name System (DNS) discovery feature of Oracle Internet Directory. Also note that if you store CRLs in the directory, then you must use the `orapki` utility to periodically update them. See Section 11.2.5.2.2.

3. CRL DP

    If the CA specifies a location in the CRL DP X.509, version 3, certificate extension when the certificate is issued, then the appropriate CRL that contains revocation information for that certificate is downloaded. Currently, Oracle Advanced Security supports downloading CRLs over HTTP and LDAP.

    > **Notes:**
    >
    > ■ For performance reasons, only user certificates are checked.
    >
    > ■ Oracle recommends that you store CRLs in the directory rather than the local file system.

11.2.5.2 Certificate Revocation List Management

Before you can enable certificate revocation status checking, you must ensure that the CRLs you receive from the CAs you use are in a form (renamed with a hash value) or in a location (uploaded to the directory) in which your system can use them. Oracle Advanced Security provides a command-line utility, `orapki`, that you can use to perform the following tasks:

- Renaming CRLs with a Hash Value for Certificate Validation
- Uploading CRLs to Oracle Internet Directory
- Listing CRLs Stored in Oracle Internet Directory
- Viewing CRLs in Oracle Internet Directory
- Deleting CRLs from Oracle Internet Directory

> **Note:** CRLs must be updated at regular intervals (before they expire) for successful validation. You can automate this task by using `orapki` commands in a script.

You can also use LDAP command-line tools to manage CRLs in Oracle Internet Directory.

> **See Also:** Appendix A, "Syntax for Command-Line Tools" in *Oracle Identity Management Application Developer's Guide* for information about LDAP command-line tools and their syntax.

11.2.5.2.1 Renaming CRLs with a Hash Value for Certificate Validation When the system validates a certificate, it must locate the CRL issued by the CA who created the certificate. The system locates the appropriate CRL by matching the issuer name in the certificate with the issuer name in the CRL.

When you specify a CRL storage location for the **Certificate Revocation Lists Path** field in Oracle Net Manager (sets the `SSL_CRL_PATH` parameter in the `sqlnet.ora` file), use the `orapki` utility to rename CRLs with a hash value that represents the issuer's name. Creating the hash value enables the server to load the CRLs.

On UNIX systems, `orapki` creates a symbolic link to the CRL. On Windows systems, it creates a copy of the CRL file. In either case, the symbolic link or the copy created by `orapki` are named with a hash value of the issuer's name. Then when the system validates a certificate, the same hash function is used to calculate the link (or copy) name so the appropriate CRL can be loaded.

Depending on your operating system, enter one of the following commands to rename CRLs stored in the file system.

To rename CRLs stored in UNIX file systems:
```
orapki crl hash -crl crl_filename [-wallet wallet_location]
-symlink crl_directory [-summary]
```

To rename CRLs stored in Windows file systems:
```
orapki crl hash -crl crl_filename
[-wallet wallet_location] -copy crl_directory [-summary]
```

In the preceding commands, `crl_filename` is the name of the CRL file, `wallet_location` is the location of a wallet that contains the certificate of the CA that issued the CRL, and `crl_directory` is the directory in which the CRL is located.

Using `-wallet` and `-summary` are optional. Specifying `-wallet` causes the tool to verify the validity of the CRL against the CA's certificate prior to renaming the CRL. Specifying the `-summary` option causes the tool to display the CRL issuer's name.

11.2.5.2.2 Uploading CRLs to Oracle Internet Directory Publishing CRLs in the directory enables CRL validation throughout your enterprise, eliminating the need for

individual applications to configure their own CRLs. All applications can use the CRLs stored in the directory in which they can be centrally managed, greatly reducing the administrative overhead of CRL management and use.

The user who uploads CRLs to the directory by using `orapki` must be a member of the directory group CRLAdmins (`cn=CRLAdmins,cn=groups,%s_ OracleContextDN%`). This is a privileged operation because these CRLs are accessible to the entire enterprise. Contact your directory administrator to be added to this administrative directory group.

To upload CRLs to the directory, enter the following at the command line:

```
orapki crl upload -crl crl_location
-ldap hostname:ssl_port -user username [-wallet wallet_location] [-summary]
```

In the preceding command, `crl_location` is the file name or URL in which the CRL is located, `hostname` and `ssl_port` (SSL port with no authentication) are for the system on which your directory is installed, `username` is the directory user who has permission to add CRLs to the CRL subtree, and `wallet_location` is the location of a wallet that contains the certificate of the CA that issued the CRL.

Using `-wallet` and `-summary` are optional. Specifying `-wallet` causes the tool to verify the validity of the CRL against the CA's certificate prior to uploading it to the directory. Specifying the `-summary` option causes the tool to print the CRL issuer's name and the LDAP entry in which the CRL is stored in the directory.

> **Note:**
>
> ■ The `orapki` utility will prompt you for the directory password when you perform this operation.
>
> ■ Ensure that you specify the directory SSL port on which the Diffie-Hellman-based SSL server is running. This is the SSL port that does not perform authentication. Neither the server authentication nor the mutual authentication SSL ports are supported by the `orapki` utility.

11.2.5.2.3 Listing CRLs Stored in Oracle Internet Directory You can display a list of all CRLs stored in the directory with `orapki`, which is useful for browsing to locate a particular CRL to view or download to your local system. This command displays the CA who issued the CRL (Issuer) and its location (DN) in the CRL subtree of your directory.

To list CRLs in Oracle Internet Directory, enter the following at the command line:

```
orapki crl list -ldap hostname:ssl_port
```

In the preceding command, the `hostname` and `ssl_port` are for the system on which your directory is installed. Note that this is the directory SSL port with no authentication as described in the preceding section.

11.2.5.2.4 Viewing CRLs in Oracle Internet Directory You can view specific CRLs that are stored in Oracle Internet Directory in a summarized format or you can request a complete listing of revoked certificates for the specified CRL. A summary listing provides the CRL issuer's name and its validity period. A complete listing provides a list of all revoked certificates contained in the CRL.

To view a summary listing of a CRL in Oracle Internet Directory, enter the following at the command line:

```
orapki crl display -crl crl_location [-wallet wallet_location] -summary
```

In the preceding command, `crl_location` is the location of the CRL in the directory. It is convenient to paste the CRL location from the list that displays when you use the `orapki crl list` command. See "Section 11.2.5.2.3.

To view a list of all revoked certificates contained in a specified CRL, which is stored in Oracle Internet Directory, enter the following at the command line:

```
orapki crl display -crl crl_location [-wallet wallet_location] -complete
```

For example, the following `orapki` command:

```
orapki crl display -crl $T_WORK/pki/wlt_crl/nzcrl.txt -wallet $T_WORK/pki/wlt_crl
-complete
```

produces the following output, which lists the CRL issuer's DN, its publication date, date of its next update, and the revoked certificates it contains:

```
issuer = CN=root,C=us, thisUpdate = Sun Nov 16 10:56:58 PST 2003,
nextUpdate = Mon Sep 30 11:56:58 PDT 2013, revokedCertificates =
{(serialNo = 153328337133459399575438325845117876415,
revocationDate - Sun Nov 16 10:56:58 PST 2003)}
CRL is valid
```

Using the `-wallet` option causes the `orapki crl display` command to validate the CRL against the CA's certificate.

Depending on the size of your CRL, choosing the `-complete` option may take a long time to display.

You can also use Oracle Directory Manager, a graphical user interface tool that is provided with Oracle Internet Directory, to view CRLs in the directory. CRLs are stored in the following directory location:

```
cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext
```

11.2.5.2.5 Deleting CRLs from Oracle Internet Directory The user who deletes CRLs from the directory by using `orapki` must be a member of the directory group `CRLAdmins`. See Section 11.2.5.2.2 for information about this directory administrative group.

To delete CRLs from the directory, enter the following at the command line:

```
orapki crl delete -issuer issuer_name -ldap hostname:ssl_port
-user username [-summary]
```

In the preceding command, `issuer_name` is the name of the CA who issued the CRL, the `hostname` and `ssl_port` are for the system on which your directory is installed, and `username` is the directory user who has permission to delete CRLs from the CRL subtree. Note that this must be a directory SSL port with no authentication. See Section 11.2.5.2.2 for more information about this port.

Using the `-summary` option causes the tool to print the CRL LDAP entry that was deleted.

For example, the following `orapki` command:

```
orapki crl delete -issuer "CN=root,C=us"
-ldap machine1:3500 -user cn=orcladmin -summary
```

produces the following output, which lists the location of the deleted CRL in the directory:

```
Deleted CRL at cn=root
cd45860c.rN,cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext
```

11.2.6 orapki Utility Commands Summary

This section lists and describes the following orapki commands:

11.2.6.1 orapki cert create

The following sections describe this command.

11.2.6.1.1 Purpose Use this command to create a signed certificate for testing purposes.

11.2.6.1.2 Syntax orapki cert create [-wallet *wallet_location*] -request *certificate_request_location* -cert *certificate_location* -validity *number_of_days* [-summary]

- The -wallet parameter specifies the wallet containing the user certificate and private key that will be used to sign the certificate request.

- The -request parameter (mandatory) specifies the location of the certificate request for the certificate you are creating.

- The -cert parameter (mandatory) specifies the directory location in which the tool places the new signed certificate.

- The -validity parameter (mandatory) specifies the number of days, starting from the current date, that this certificate will be valid.

11.2.6.2 orapki cert display

The following sections describe this command.

11.2.6.2.1 Purpose Use this command to display details of a specific certificate.

11.2.6.2.2 Syntax orapki cert display -cert *certificate_location* [-summary|-complete]

- The `-cert` parameter specifies the location of the certificate you want to display.

- You can use either the `-summary` or the `-complete` parameter to display the following information:

  - `-summary` displays the certificate and its expiration date

  - `-complete` displays additional certificate information, including the serial number and public key

11.2.6.3 orapki crl delete

The following sections describe this command.

11.2.6.3.1 Purpose Use this command to delete CRLs from Oracle Internet Directory. Note that the user who deletes CRLs from the directory by using `orapki` must be a member of the `CRLAdmins` (`cn=CRLAdmins,cn=groups,%s_OracleContextDN%`) directory group.

11.2.6.3.2 Syntax `orapki crl delete -issuer` *issuer_name* `-ldap` *hostname*:*ssl_port* `-user` *username* `[-summary]`

- The `-issuer` parameter specifies the name of the certificate authority (CA) who issued the CRL.

- The `-ldap` parameter specifies the hostname and SSL port for the directory in which the CRLs are to be deleted. Note that this must be a directory SSL port with no authentication. See Section 11.2.5.2.2 for more information about this port.

- The `-user` parameter specifies the username of the directory user who has permission to delete CRLs from the CRL subtree in the directory.

- The `-summary` parameter is optional. Using it causes the tool to print the CRL LDAP entry that was deleted.

11.2.6.4 orapki crl display

The following sections describe this command.

11.2.6.4.1 Purpose Use this command to display specific CRLs that are stored in Oracle Internet Directory.

11.2.6.4.2 Syntax `orapki crl display -crl` *crl_location* `[-wallet` *wallet_location*`] [-summary|-complete]`

- The `-crl` parameter specifies the location of the CRL in the directory. It is convenient to paste the CRL location from the list that displays when you use the `orapki crl list` command. See Section 11.2.6.6.

- The `-wallet` parameter (optional) specifies the location of the wallet that contains the certificate of the certificate authority (CA) who issued the CRL. Using it causes the tool to verify the validity of the CRL against the CA's certificate prior to displaying it.

- Choosing either the -summary or the -complete parameters displays the following information:

  - `-summary` provides a listing that contains the CRL issuer's name and the CRL's validity period

– `-complete` provides a list of all revoked certificates that the CRL contains. Note that this option may take a long time to display, depending on the size of the CRL.

11.2.6.5 orapki crl hash

The following sections describe this command.

11.2.6.5.1 Purpose Use this command to generate a hash value of the certificate revocation list (CRL) issuer to identify the location of the CRL in your file system for certificate validation.

11.2.6.5.2 Syntax `orapki crl hash -crl crl_filename|URL` `[-wallet wallet_location] [-symlink|-copy] crl_directory [-summary]`

- The `-crl` parameter specifies the filename that contains the CRL or the URL in which it can be found.

- The `-wallet` parameter (optional) specifies the location of the wallet that contains the certificate of the certificate authority (CA) who issued the CRL. Using it causes the tool to verify the validity of the CRL against the CA's certificate prior to uploading it to the directory.

- Depending on your operating system, use either the `-symlink` or the `-copy` parameter:

  – On UNIX: Use `-symlink` to create a symbolic link to the CRL at the `crl_directory` location

  – On Windows: Use `-copy` to create a copy of the CRL at the `crl_directory` location

- The `-summary` parameter (optional) causes the tool to display the CRL issuer's name.

11.2.6.6 orapki crl list

The following sections describe this command.

11.2.6.6.1 Purpose Use this command to display a list of CRLs stored in Oracle Internet Directory. This is useful for browsing to locate a particular CRL to view or download to your local file system.

11.2.6.6.2 Syntax `orapki crl list -ldap hostname:ssl_port`

The `-ldap` parameter specifies the hostname and SSL port for the directory server from which you want to list CRLs. Note that this must be a directory SSL port with no authentication. See Section 11.2.5.2.2 for more information about this port.

11.2.6.7 orapki crl upload

The following sections describe this command.

11.2.6.7.1 Purpose Use this command to upload certificate revocation lists (CRLs) to the CRL subtree in Oracle Internet Directory. Note that you must be a member of the directory administrative group CRLAdmins (`cn=CRLAdmins,cn=groups,%s_ OracleContextDN%`) to upload CRLs to the directory.

11.2.6.7.2 Syntax `orapki crl upload -crl crl_location`

```
-ldap hostname:ssl_port -user username
[-wallet wallet_location] [-summary]
```

- The `-crl` parameter specifies the directory location or the URL of the CRL that you are uploading to the directory.

- The `-ldap` parameter specifies the hostname and SSL port for the directory to which you are uploading the CRLs. Note that this must be a directory SSL port with no authentication. See Section 11.2.5.2.2 for more information about this port.

- The `-user` parameter specifies the username of the directory user who has permission to add CRLs to the CRL subtree in the directory.

- The `-wallet` parameter specifies the location of the wallet that contains the certificate of the certificate authority (CA) who issued the CRL. This is an optional parameter. Using it causes the tool to verify the validity of the CRL against the CA's certificate prior to uploading it to the directory.

- The `-summary` parameter is also optional. Using it causes the tool to display the CRL issuer's name and the LDAP entry in which the CRL is stored in the directory.

11.2.6.8 orapki wallet add

The following sections describe this command.

11.2.6.8.1 Purpose Use this command to add certificate requests and certificates to an Oracle wallet.

11.2.6.8.2 Syntax **To add certificate requests:**

```
orapki wallet add -wallet wallet_location -dn user_dn -keySize 512|1024|2048
```

- The `-wallet` parameter specifies the location of the wallet to which you want to add a certificate request.

- The `-dn` parameter specifies the distinguished name of the certificate owner.

- The `-keySize` parameter specifies the key size for the certificate.

- To sign the request, export it with the export option. See Section 11.2.6.11.

To add trusted certificates:

```
orapki wallet add -wallet wallet_location -trusted_cert -cert certificate_location
```

- The `-trusted_cert` parameter causes the tool to add the trusted certificate, at the location specified with `-cert`, to the wallet.

To add root certificates:

```
orapki wallet add -wallet wallet_location -dn
certificate_dn -keySize 512|1024|2048 -self_signed -validity number_of_days
```

- The `-self_signed` parameter causes the tool to create a root certificate.

- The `-validity` parameter is mandatory. Use it to specify the number of days, starting from the current date, that this root certificate will be valid.

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.

11.2.6.9 orapki wallet create

The following sections describe this command.

11.2.6.9.1 Purpose Use this command to create an Oracle wallet or to set auto login on for an Oracle wallet.

11.2.6.9.2 Syntax orapki wallet create -wallet *wallet_location* [-auto_login]

- The -wallet parameter specifies a location for the new wallet or the location of the wallet for which you want to turn on auto login.
- The -auto_login parameter creates an auto login wallet, or it turns on automatic login for the wallet specified with the -wallet option. See Section 11.1.4.14 for details about auto login wallets.

11.2.6.10 orapki wallet display

The following sections describe this command.

11.2.6.10.1 Purpose Use this command to view the certificate requests, user certificates, and trusted certificates in an Oracle wallet.

11.2.6.10.2 Syntax orapki wallet display -wallet *wallet_location*

- The -wallet parameter specifies a location for the wallet you want to open if it is not located in the current working directory.

11.2.6.11 orapki wallet export

The following sections describe this command.

11.2.6.11.1 Purpose Use this command to export certificate requests and certificates from an Oracle wallet.

11.2.6.11.2 Syntax orapki wallet export -wallet *wallet_location* -dn *certificate_dn* -cert *certificate_filename*

- The -wallet parameter specifies the location of the wallet from which you want to export the certificate.
- The -dn parameter specifies the distinguished name of the certificate.
- The -cert parameter specifies the name 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 name of the file that contains the exported certificate request.

11.3 Interoperability with X.509 Certificates

Oracle Wallet Manager functionality supports users who already have certificates provisioned. If you do not use Oracle Wallet Manager to create certificates, you can use it to manage and store certificates created previously.

11.3.1 Public-Key Cryptography Standards (PKCS) Support

Oracle Wallet Manager stores X.509 certificates and private key**s** in Public-Key Cryptography Standards (PKCS) #12 format, and generates certificate requests according to the PKCS #10 specification developed by RSA Laboratories. This makes the Oracle wallet structure interoperable with supported third party PKI applications, and provides wallet portability across operating systems.

Oracle Wallet Manager wallets can be enabled to store credentials on hardware security modules using APIs that conform to the PKCS #11 specification. When `PKCS11` wallet type is chosen at the time of wallet creation, then all keys stored in that wallet are saved to a hardware security module or token, such as smart cards, PCMCIA cards, smart diskettes, or other types of portable hardware devices that store private keys, perform cryptographic operations, or both.

> **See Also:**
>
> - Section 11.1.5.1.3, "Importing Certificates Created with a Third-Party Tool"
>
> - Section 11.1.4.5, "Exporting Oracle Wallets to Third-Party Environments"
>
> - Section 11.1.4.2.2, "Creating a Wallet to Store Hardware Security Module Credentials"
>
> - To view PKCS standards documents, navigate to the following URL:
>
>   http://www.rsasecurity.com/rsalabs/

11.3.2 Multiple Certificate Support

Oracle Wallet Manager enables you to store multiple certificate**s** for each wallet, supporting the following Oracle PKI certificate usages:

- SSL

- S/MIME signature

- S/MIME encryption

- Code-Signing

- CA Certificate Signing

Oracle Wallet Manager supports multiple certificates for a single digital entity, where each certificate can be used for a set of Oracle PKI certificate usages, but the same certificate cannot be used for all such usages (See Table 11–4 and Table 11–5 for legal usage combinations). There must be a one-to-one mapping between certificate requests and certificates. The same certificate request can be used to obtain multiple certificates; however, more than one certificate for each certificate request cannot be installed in the same wallet at the same time.

Oracle Wallet Manager uses the X.509 Version 3 `KeyUsage` extension types to define Oracle PKI certificate usages. The key usage extension types are optional bits that can

be set in certificates. Setting these bits defines what purpose the certificate's key can be used for. When certificates are issued, the certificate authority sets these bits according to the type of certificate that you have requested. Table 11–4 lists and describes these key usage types.

Table 11–4   X.509 Version 3 KeyUsage Extension Types, Values, and Descriptions

KeyUsage Extension Type	Value	Description
digitalSignature	0	Used for entity authentication and to authenticate data origin integrity.
nonRepudiation	1	Used to protect against the signing entity falsely denying some action.
keyEncipherment	2	Used when the subject public key is used for key transport.
dataEncipherment	3	Used when the subject public key is used for enciphering data, other than cryptographic keys.
keyAgreement	4	Used when the subject public key is used for key agreement during SSL connection negotiation.
keyCertSign	5	Used when the subject public key is used for verifying a signature on certificates. May only be used in CA certificates.
cRLSign	6	Used when the subject public key is used for verifying a signature on certificate revocation lists.
encipherOnly	7	When the encipherOnly bit is asserted, the keyAgreement bit must also be set. When these two bits are set the subject public key may be used only for enciphering data while performing key agreement.
decipherOnly	8	As with the encipherOnly bit, the keyAgreement bit must also be set when decipherOnly is set. When these two bits (decipherOnly and keyAgreement) are set the subject public key may be used only for deciphering data while performing key agreement.

> **See Also:**   The Internet Engineering Task Force RFC #2459, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, for a complete description of the `KeyUsage` extension types at the following URL:
>
> http://www.ietf.org/rfc/rfc2459.txt

When installing a certificate (user certificate or trusted certificate), Oracle Wallet Manager maps the `KeyUsage` extension values to Oracle PKI certificate usages as specified in Table 11–4 and Table 11–5.

Table 11–5   Oracle Wallet Manager Import of Trusted Certificates to an Oracle Wallet

KeyUsage Value	Critical?[1]	Usage
none	NA	Importable.
Any combination excluding 5	Yes	Not importable.
Any combination excluding 5	No	Importable.
5 alone, or any combination including 5	NA	Importable.

[1]   If the `KeyUsage` extension is *critical*, the certificate cannot be used for other purposes.

You should obtain certificates from the certificate authority with the correct `KeyUsage` value for the required Oracle PKI certificate usage. A single wallet can contain multiple key pairs for the same usage. Each certificate can support multiple Oracle PKI certificate usages, as indicated by Table 11–4 and Table 11–5. Oracle PKI applications use the first certificate containing the required PKI certificate usage.

For example, for SSL usage, the first certificate containing the SSL Oracle PKI certificate usage is used.

If you do not have a certificate with SSL usage, then an `ORA-28885` error (`No certificate with required key usage found`) is returned.

12

Enabling SSL in the Infrastructure

This chapter provides instructions for enabling SSL in Oracle Application Server Infrastructure installations.

> **Note:** In this chapter, references to any of the following Oracle Application Server products are applicable for Release 10.1.4, Release 2 (10.1.2), or earlier software only:
>
> - OracleAS Infrastructure
> - OracleAS Single Sign-On
> - OracleAS Web Cache
> - OracleAS Certificate Authority
> - Oracle Delegated Administration Services
> - Oracle Identity Management

The following topics are covered:

- SSL Communication Paths in the Infrastructure
- Recommended SSL Configurations
- Common SSL Configuration Tasks

12.1 SSL Communication Paths in the Infrastructure

This section identifies all SSL communication paths used in the Oracle Application Server Infrastructure, and provides cross-references to the configuration instructions in component guides in the Oracle Application Server documentation library.

> **Note:** When you install Oracle Identity Management, you are prompted to select a mode for Oracle Internet Directory. The default mode is dual mode, which allows some components to access Oracle Internet Directory using non-SSL connections. If SSL mode was chosen during installation, then all installed components must use SSL when connecting to the directory.
>
> Before you begin SSL configuration, determine the Oracle Internet Directory mode. Start the `oidadmin` tool and view the SSL mode in Oracle Directory Manager. Go to the Directory Server and select **View Properties > SSL Settings**.

The following are the communication paths through the Oracle Application Server Infrastructure, and their related SSL configuration instructions:

- **Oracle HTTP Server to the OC4J_SECURITY instance**

  To configure the AJP communication over SSL, you must configure mod_oc4j's communication with the `iaspt` daemon. To do this, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Configuring mod_oc4j to Use SSL."

- **Oracle HTTP Server to iaspt (Port Tunneling) and then to the OC4J_SECURITY instance**

  To configure this connection path for SSL, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Understanding Port Tunneling."

- **The OC4J_SECURITY instance to Oracle Internet Directory**

  To configure this connection path for SSL, follow the instructions in the *Oracle Application Server Single Sign-On Administrator's Guide*, section titled "Enable SSL on the Single Sign-On Middle Tier." It explains how to configure SSL communication between the browser and the OracleAS Single Sign-On server.

  Oracle Delegated Administration Services is SSL-enabled after you configure the Oracle HTTP Server for SSL. The Oracle Delegated Administration Services communication to Oracle Internet Directory is always SSL-enabled; you do not have to perform any configuration tasks to accomplish this. (OracleAS Single Sign-On, Oracle Application Server Certificate Authority, and Oracle Delegated Administration Services communicate with Oracle Internet Directory in SSL mode by default.)

- **Oracle Directory Integration and Provisioning to Oracle Internet Directory and Oracle Internet Directory replication server to Oracle Internet Directory**

  As shown in Figure 12–1, a variety of components and communication paths may be configured for SSL. The following lists references to the instructions for each:

  - Communication between the Oracle Internet Directory Replication server and the Oracle Internet Directory server: *Oracle Application Server High Availability Guide*, section titled "Secure Sockets Layer (SSL) and Oracle Internet Directory Replication"

  - Communication between Oracle Directory Integration and Provisioning and the Oracle Internet Directory server: *Oracle Identity Management Integration Guide*, chapter titled "Oracle Directory Integration and Provisioning Server Administration"

- **The OC4J_SECURITY instance to the Metadata Repository database and Oracle Internet Directory to the Metadata Repository database**

  If Oracle Internet Directory is configured to accept SSL connections on the SSL port specified, you need only specify the SSL protocol and SSL port in the JDBC URL requesting an application, as follows:

  ldaps://*host:sslport*/...

  Note that when you are using a secure connection, you must add an s to the name of the protocol. For example, use ldaps instead of ldap.

  If Oracle Internet Directory is not configured to accept SSL connections on the SSL port, you must modify the configuration. See *Oracle Internet Directory Administrator's Guide*, section titled "Secure Sockets Layer (SSL) and the Directory."

Figure 12–1   Oracle Identity Management Components and SSL Connection Paths
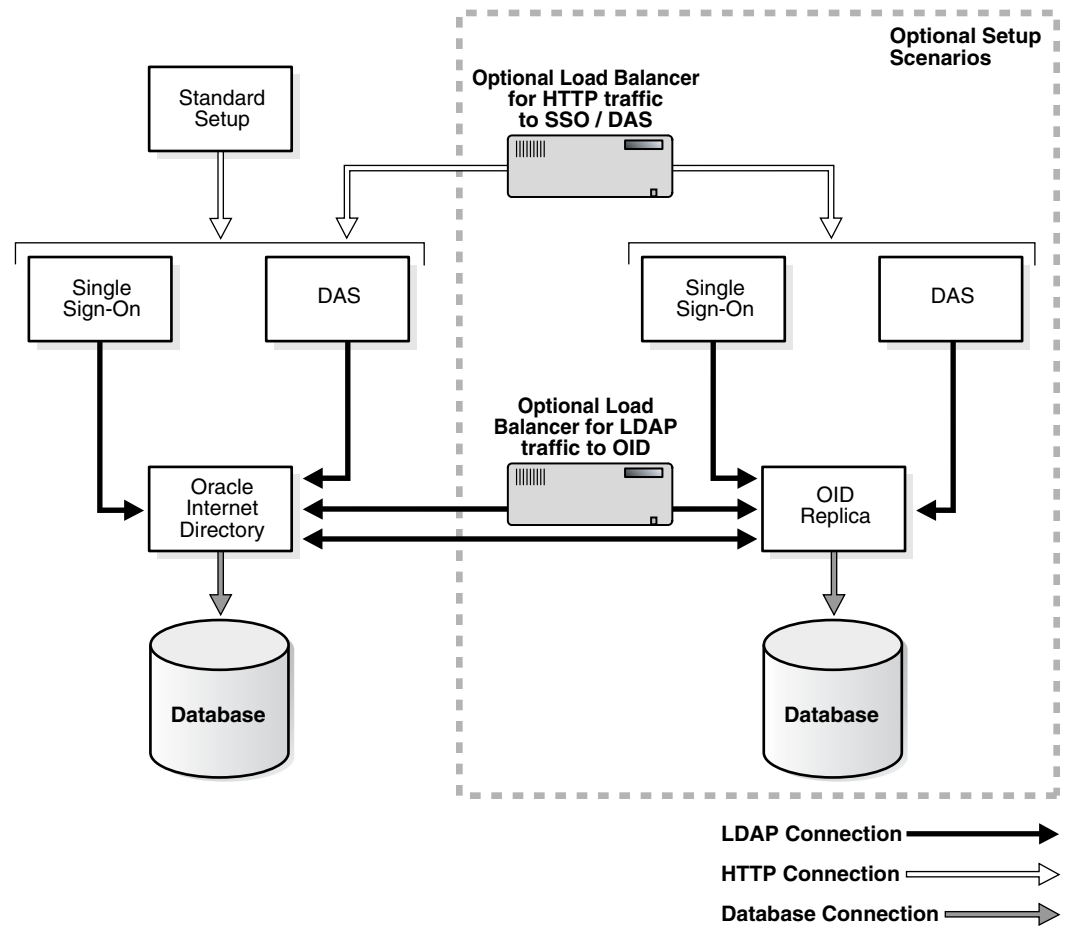
12.2 Recommended SSL Configurations

The *Oracle Application Server Security Guide* discusses security concepts in detail and provides recommendations for configuring security in various configurations. The "Recommended Deployment Topologies" chapter presents sample architectures. After you have identified the components on which you need to enable SSL, use the instructions in this chapter and Chapter 13, "Enabling SSL in the Middle Tier" to configure the components.

Configuring SSL in OracleAS Single Sign-On and Oracle Delegated Administration Services is typical in the recommended deployment topologies (as described in Section 12.3.1). Configuring SSL in all Infrastructure communication paths is described in Section 12.1.

12.3 Common SSL Configuration Tasks

This section provides references to the component guides in the Oracle Application Server documentation library that provide instructions for configuring SSL in individual components.

12.3.1 Configuring SSL for OracleAS Single Sign-On and Oracle Delegated Administration Services

Follow the instructions in the *Oracle Application Server Single Sign-On Administrator's Guide* to configure SSL communication between:

- The browser and the OracleAS Single Sign-On server (section titled "Enable SSL on the Single Sign-On Middle Tier")

- The OracleAS Single Sign-On server and the Oracle Internet Directory server (section titled "Configuring SSL Between the Single Sign-On Server and Oracle Internet Directory")

Oracle Delegated Administration Services is SSL-enabled after you configure the Oracle HTTP Server for SSL (as described in "Enable SSL on the Single Sign-On Middle Tier"). The Oracle Delegated Administration Services communication to Oracle Internet Directory is always SSL-enabled; you do not have to perform any configuration tasks to accomplish this.

12.3.2 Configuring SSL for Oracle Internet Directory

Instructions for configuring SSL communication in Oracle Internet Directory are provided in the following:

- *Oracle Internet Directory Administrator's Guide*, section titled "Secure Sockets Layer (SSL) and the Directory"

- *Oracle Internet Directory Administrator's Guide*, section titled "Configuring SSL Parameters"

- *Oracle Internet Directory Administrator's Guide*, section titled "Limitations of the Use of SSL in 10*g* (10.1.2)"

12.3.3 Configuring SSL for Oracle Internet Directory Replication Server and Oracle Directory Integration and Provisioning

As shown in Figure 12–1, a variety of components and communication paths may be configured for SSL. The following lists references to the instructions for each:

- Communication between the Oracle Internet Directory Replication server and the Oracle Internet Directory server: *Oracle Application Server High Availability Guide*, section titled "Secure Sockets Layer (SSL) and Oracle Internet Directory Replication"

- Communication between Oracle Directory Integration and Provisioning and the Oracle Internet Directory server: *Oracle Identity Management Integration Guide*, chapter titled "Oracle Directory Integration and Provisioning Server Administration"

12.3.4 Configuring SSL in the Identity Management Database

Follow the instructions in the *Oracle Application Server Single Sign-On Administrator's Guide*, section titled "Reconfigure the Identity Management Infrastructure Database" to configure SSL communication to the Identity Management database.

12.3.5 Additional SSL Configuration in the OC4J_SECURITY Instance

This section provides references to SSL configuration information for mod_oc4j and OC4J.

12.3.5.1 Configuring SSL from mod_oc4j to OC4J_SECURITY

To configure the AJP communication over SSL, you must configure mod_oc4j's communication with the `iaspt` daemon. To do this, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Enabling SSL between mod_oc4j and OC4J."

12.3.5.2 Using Port Tunneling from mod_oc4j to the OC4J_SECURITY Instance

To configure this connection path for SSL, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Understanding Port Tunneling."

12.3.5.3 Configuring JDBC/SSL (ASO support)

If Oracle Internet Directory is configured to accept SSL connections on the SSL port specified, you need only specify the SSL protocol and SSL port in the JDBC URL requesting an application, as follows:

```
ldaps://host:sslport/...
```

Note that when you are using a secure connection, you must add an "s" to the name of the protocol (for example, use `ldaps` instead of `ldap`).

If Oracle Internet Directory is not configured to accept SSL connections on the SSL port, you must modify the configuration. See *Oracle Internet Directory Administrator's Guide*, section titled "Secure Sockets Layer (SSL) and the Directory."

12.3.6 SSL in Oracle Application Server Certificate Authority

Oracle Application Server Certificate Authority is SSL-enabled by default, so there are no configuration tasks associated with this component.

> **Tip:** OracleAS Certificate Authority simplifies the task of certificate provisioning for Oracle Identity Management users (certificates are automatically provisioned to SSO-authenticated users).
>
> To enable certificate-based authentication using OCA OracleAS Single Sign-On, see the *Oracle Application Server Certificate Authority Administrator's Guide*. To enable certificate-based authentication to OracleAS Single Sign-On, see the *Oracle Application Server Single Sign-On Administrator's Guide*.

12.3.7 Configuring SSL for Oracle Enterprise Manager 10*g*

Oracle Enterprise Manager 10*g* comprises two components, each of which can be configured for SSL communication: Grid Control Console and Application Server Control.

12.3.7.1 Configuring Security for the Grid Control

Follow the steps in the "Configuring Security (SSL and HTTPS) for Grid Control" in the *Oracle Enterprise Manager Advanced Configuration Guide*.

12.3.7.2 Configuring Security for Application Server Control

The communication paths of interest in the SSL configuration of Application Server Control are shown in Figure 12–2, and are labeled **https**.

Figure 12–2   SSL Connection Paths in Oracle Enterprise Manager 10g
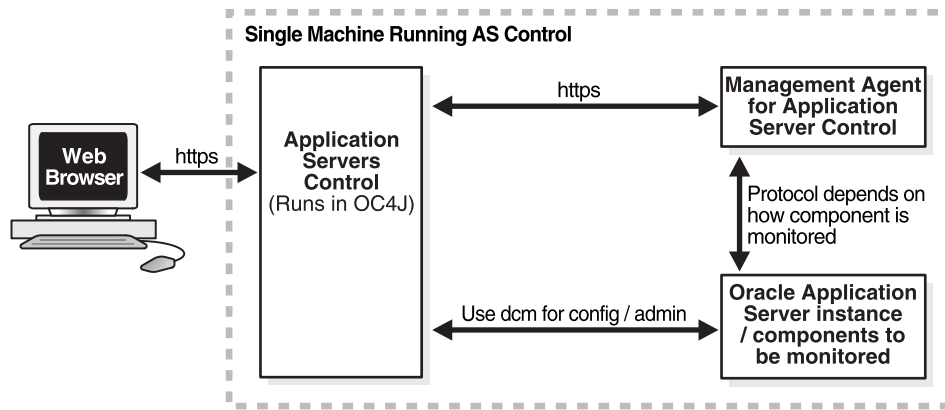
To secure the communications between the Web browser and Application Server Control, and between Application Server Control and the Oracle Application Server components, refer to Section A.3.

13

Enabling SSL in the Middle Tier

This chapter provides instructions for enabling SSL in Oracle Application Server middle-tier installations.

> **Note:** In this chapter, references to any of the following Oracle Application Server products are applicable for Release 10.1.4, Release 2 (10.1.2) or earlier software only:
>
> - OracleAS Single Sign-On
> - OracleAS Web Cache
> - Oracle Internet Directory

The following topics are covered:

- SSL Communication Paths in the Middle Tier
- Recommended SSL Configurations
- Common SSL Configuration Tasks for the Middle Tier

13.1 SSL Communication Paths in the Middle Tier

This section identifies all SSL communication paths used in the Oracle Application Server middle-tier installation types, and provides cross-references to the configuration instructions in component guides in the Oracle Application Server documentation library.

The following are communication paths through the Oracle Application Server middle tier, and their related SSL configuration instructions:

- **External Clients or Load Balancer to Oracle HTTP Server**

  To configure the Oracle HTTP Server for SSL, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Enabling SSL."

- **External Clients or Load Balancer to OracleAS Web Cache**

  To configure OracleAS Web Cache for SSL, follow the instructions in "Configuring OracleAS Web Cache for HTTPS Requests" in the *Oracle Application Server Web Cache Administrator's Guide*.

- **OracleAS Web Cache to Oracle HTTP Server**

  To configure OracleAS Web Cache for SSL, follow the instructions in "Configuring OracleAS Web Cache for HTTPS Requests" in the *Oracle Application Server Web Cache Administrator's Guide*.

- **Oracle HTTP Server to OC4J Applications (AJP)**

  To configure the AJP communication over SSL, you must configure mod_oc4j's communication with the `iaspt` daemon. To do this, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Configuring mod_oc4j to Use SSL."

- **Oracle HTTP Server to iaspt and then to OC4J**

  To configure this connection path for SSL, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Understanding Port Tunneling."

- **OC4J (the JAAS provider) to Oracle Internet Directory**

  To configure the provider, follow the instructions in the *Oracle Containers for J2EE Security Guide*. To configure the provider for SSL, set the `SSL_ONLY_FLAG` to `true`.

- **OC4J to the database (ASO)**

  If Oracle Internet Directory is configured to accept SSL connections on the SSL port specified, you need only specify the SSL protocol and SSL port in the JDBC URL requesting an application, as follows:

  ```
  ldaps://host.sslport/...
  ```

  Note that when you are using a secure connection, you must add an s to the name of the protocol. For example, use ldaps instead of ldap.

  If Oracle Internet Directory is not configured to accept SSL connections on the SSL port, you must modify the configuration. See *Oracle Internet Directory Administrator's Guide*, section titled "Secure Sockets Layer (SSL) and the Directory."

- **ORMI (Oracle Remote Method Invocation, a custom wire protocol) over SSL**

  To configure this connection path for SSL, refer to the *Oracle Containers for J2EE Security Guide*.

- **SSL into standalone OC4J (HTTPS)**

  To configure this connection path for SSL, follow the instructions in the *Oracle Containers for J2EE Security Guide*, section titled "Configuring SSL in OC4J." It explains how to use SSL to secure communication between clients and an OC4J instance.

- **OracleAS Portal Parallel Page Engine (the servlet in the OC4J_PORTAL instance) to OracleAS Web Cache (HTTPS)**

  To configure this connection path for SSL, follow the instructions in the *Oracle Containers for J2EE Security Guide*, section titled "Configuring SSL in OC4J."

13.2 Recommended SSL Configurations

The *Oracle Application Server Security Guide* discusses security concepts in detail and provides recommendations for configuring security in various configurations. The "Recommended Deployment Topologies" chapter presents sample architectures for installation types. After you have identified the components on which you need to enable SSL, use the instructions in this chapter and Chapter 12, "Enabling SSL in the Infrastructure" to configure the components.

13.3 Common SSL Configuration Tasks for the Middle Tier

This section identifies some commonly used SSL configurations in the Oracle Application Server middle-tier installation types, and provides cross-references to the configuration instructions in component guides in the Oracle Application Server documentation library.

13.3.1 Enabling SSL in OracleAS Web Cache

OracleAS Web Cache is part of Oracle Application Server middle-tier installations. To configure it for SSL, follow the instructions in the chapter "Configuring OracleAS Web Cache for HTTPS Requests" in the *Oracle Application Server Web Cache Administrator's Guide*.

13.3.2 Enabling SSL in the Oracle HTTP Server

Oracle HTTP Server is part of all Oracle Application Server middle-tier installations. To configure it for SSL, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Enabling SSL."

13.3.3 Enabling SSL in OC4J

To configure SSL connections to OC4J clients, follow the instructions in the *Oracle Containers for J2EE Security Guide*, section titled "Oracle HTTPS for Client Connections."

13.3.3.1 Configuring SSL from Oracle HTTP Server to OC4J

To configure the AJP communication over SSL, you must configure `mod_oc4j`'s communication with the `iaspt` daemon. To do this, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Enabling SSL between mod_oc4j and OC4J."

13.3.3.2 Using Port Tunneling (iaspt) from Oracle HTTP Server to OC4J

To configure this connection path for SSL, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Understanding Port Tunneling."

13.3.3.3 Configuring ORMI/HTTP SSL

ORMI over SSL is not supported. To configure similar functionality, you can configure ORMI over HTTP, and then configure HTTP for SSL.

See the *Oracle Containers for J2EE Services Guide*, section titled "Configuring ORMI Tunnelling Through HTTP" for instructions on how to configure ORMI/HTTP.

13.3.3.4 Configuring the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider for SSL with Oracle Internet Directory

To configure the provider, follow the instructions in the *Oracle Application Server Enterprise Deployment Guide*, section titled "Configuring Application Authentication and Authorization." To configure the provider for SSL, set the `SSL_ONLY_FLAG` to `true`.

13.3.3.5 Configuring Oracle HTTP Server for SSL

The *Oracle Containers for J2EE Security Guide*, section titled "Enabling SSL in OC4J" explains how to configure Oracle HTTP Server for SSL.

13.3.3.6 Configuring SSL in Standalone OC4J Installations

The *Oracle Containers for J2EE Security Guide*, section titled "Enabling SSL in OC4J" explains how to use SSL to secure communication between clients and an OC4J instance.

13.3.4 Enabling SSL in J2EE and Web Cache Installations

Depending on your security needs and the configuration of the Oracle Application Server J2EE and Web Cache installation, you may implement secure communication in one or more of the installed components. Configuring the first listener (whether it is OracleAS Web Cache or the Oracle HTTP Server) may be sufficient.

To configure the Oracle HTTP Server for SSL, follow the steps in "Enabling SSL for Oracle HTTP Server" in the *Oracle HTTP Server Administrator's Guide*.

To configure OracleAS Web Cache for SSL, follow the instructions in "Configuring OracleAS Web Cache for HTTPS Requests" in the *Oracle Application Server Web Cache Administrator's Guide*.

13.3.5 Enabling SSL in Oracle Content DB

The *Oracle Content Database for Oracle WebCenter Suite Administrator's Guide*, section titled "SSL Configuration for Oracle Content DB," explains how to enable SSL for Oracle Content DB.

13.3.6 Enabling SSL in Virtual Hosts

You can use virtual hosts to deploy multiple Web sites on a single Oracle HTTP Server (for example, to make an application available over the HTTP protocol and the HTTPS protocol).

The *Oracle Application Server Single Sign-On Administrator's Guide*, section titled "Configuring mod_osso with Virtual Hosts" contains instructions on configuring an SSL virtual host to be protected by mod_osso. You cannot use name-based virtual hosting. You must use IP-based or port-based virtual hosting.

The scenario presented assumes that the following conditions are in effect:

- The host name of the application middle tier is app.mydomain.com (replace this name with the host name of your application middle tier).

- The middle tier is already configured as a non-SSL partner application (this is typically done during installation).

- The default SSL port number of the application middle tier is 4443.

13.3.7 Configuring SSL for Oracle Enterprise Manager 10*g*

See Section 12.3.7, "Configuring SSL for Oracle Enterprise Manager 10g".

14

Troubleshooting SSL

This chapter lists common questions and errors related to SSL.

It contains these topics:

- Name-Based Virtual Hosting and SSL
- Common ORA Errors Related to SSL

14.1 Name-Based Virtual Hosting and SSL

You cannot use name-based virtual hosting with SSL. This is a limitation of SSL.

If you need to configure multiple virtual hosts with SSL, here are some possible workarounds:

- Use IP-based virtual hosting. To do this, you configure multiple IP addresses for your computer, and map each IP address to a different virtual name.

- If you are willing to use nonstandard port numbers, you can associate the same IP with different names, but you must configure each name with a different port number (for example, *name1*:443, *name2*:553). This enables you to use the same IP, but you have to use nonstandard port numbers. Only one name can use the standard 443 port; other names must use other port numbers.

14.2 Common ORA Errors Related to SSL

You may need to enable Oracle Net tracing to determine the cause of an error. For information about setting tracing parameters for Oracle Net, see *Oracle Database Net Services Administrator's Guide*.

ORA-28759: Failure to Open File

Cause: The system could not open the specified file. Typically, this error occurs because the Oracle wallet cannot be found.

Action: Check the following:

- Ensure that the Oracle wallet is located either in the default location (ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default) or in the location specified by the SSLWallet directive in the ORACLE_HOME/Apache/Apache/conf/ssl.conf file. This should be the same directory location where you saved the wallet.

- Enable Oracle Net tracing to determine the name of the file that cannot be opened and the reason.

- Ensure that auto login was enabled when you saved the Oracle wallet. See Section 11.1.4.14 for details.

ORA-28786: Decryption of Encrypted Private Key Failure

Cause: An incorrect password was used to decrypt an encrypted private key. Frequently, this happens because an auto login wallet is not being used.

Action: Use Oracle Wallet Manager to enable the auto login feature for the wallet. Then re-save the wallet. See Section 11.1.4.14.

ORA-28858: SSL Protocol Error

Cause: This is a generic error that can occur during SSL handshake negotiation between two processes.

Action: Enable Oracle Net tracing and attempt the connection again to produce trace output. Then, contact Oracle customer support with the trace output.

ORA-28859 SSL Negotiation Failure

Cause: An error occurred during the negotiation between two processes as part of the SSL protocol. This error can occur when two sides of the connection do not support a common cipher suite.

Action: Ensure that the cipher suites configured on Oracle HTTP Server and on the client (which is the browser) are compatible for both client and server.

To check the cipher suites configured on Oracle HTTP Server, check the `SSLCipherSuite` directive in the `ORACLE_HOME/Apache/Apache/conf/ssl.conf` file.

To check the cipher suites configured on your browser, see the documentation for your browser. Each type of browser has its own way of setting the cipher suite.

You should also ensure that the SSL versions on both the client and the server match, or are compatible. For example, if the server accepts only SSL 3.0 and the client accepts only TLS 1.0, then the SSL connection will fail.

ORA-28862: SSL Connection Failed

Cause: This error occurred because the peer closed the connection.

Action: Check the following:

- Ensure that the Oracle wallet is located either in the default location (`ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default`) or in the location specified by the `SSLWallet` directive in the `ORACLE_HOME/Apache/Apache/conf/ssl.conf` file. This should be the same directory location where you saved the wallet.

- Check that the cipher suites are compatible for both client and server. See "ORA-28859 SSL Negotiation Failure" for details on how to check the cipher suite.

- Check that the names of the cipher suites are spelled correctly.

- Ensure that the SSL versions on both the client and the server match, or are compatible. Sometimes this error occurs because the SSL version specified on the server and client do not match. For example, if the server accepts only SSL 3.0 and the client accepts only TLS 1.0, then the SSL connection will fail.

- For more diagnostic information, enable Oracle Net tracing on the peer.

ORA-28865: SSL Connection Closed

Cause: The SSL connection closed because of an error in the underlying transport layer, or because the peer process quit unexpectedly.

Action: Check the following:

- Ensure that the SSL versions on both the client and the server match, or are compatible. Sometimes this error occurs because the SSL version specified on the server and client do not match. For example, if the server accepts only SSL 3.0 and the client accepts only TLS 1.0, then the SSL connection will fail.

- If you are using a Diffie-Hellman anonymous cipher suite and the SSLVerifyClient directive is set to require in the ssl.conf file, then the client does not pass its certificate to the server. When the server does not receive the client's certificate, the server cannot authenticate the client so the connection is closed. To resolve this, use a different cipher suite, or set the SSLVerifyClient directive to none or optional.

  See "ORA-28859 SSL Negotiation Failure" for details on how to check the cipher suite.

- Enable Oracle Net tracing and check the trace output for network errors.

ORA-28868: Peer Certificate Chain Check Failed

Cause: When the peer presented the certificate chain, it was checked and that check failed. This failure can be caused by a number of problems, including:

- One of the certificates in the chain is expired.

- A certificate authority for one of the certificates in the chain is not recognized as a trust point.

- The signature in one of the certificates cannot be verified.

Action: Follow the instructions in Section 11.1.4.3 to use Oracle Wallet Manager to open your wallet, and check the following:

- Ensure that all of the certificates installed in your wallet are current (not expired).

- Ensure that a certificate authority's certificate from your peer's certificate chain is added as a trusted certificate in your wallet. See Section 11.1.5.2.1 to use Oracle Wallet Manager to import a trusted certificate.

ORA-28885: No certificate with the required key usage found.

Cause: Your certificate was not created with the appropriate X.509 Version 3 key usage extension.

Action: Use Oracle Wallet Manager to check the certificate's key usage. See Table 11–4, " X.509 Version 3 KeyUsage Extension Types, Values, and Descriptions".

ORA-29024: Certificate Validation Failure

Cause: The certificate sent by the other side could not be validated. This may occur if the certificate has expired, has been revoked, or is invalid for another reason.

Action: Check the following:

- Check the certificate to determine whether it is valid. If necessary, get a new certificate, inform the sender that her certificate has failed, or resend.

- Check to ensure that the server's wallet has the appropriate trust points to validate the client's certificate. If it does not, then use Oracle Wallet Manager

to import the appropriate trust point into the wallet. See Section 11.1.5.2.1 for details.

- Ensure that the certificate has not been revoked and that certificate revocation list (CRL) checking is enabled. See Section 11.2.5.

ORA-29223: Cannot Create Certificate Chain

Cause: A certificate chain cannot be created with the existing trust points for the certificate being installed. Typically, this error is returned when the peer does not give the complete chain and you do not have the appropriate trust points to complete it.

Action: Use Oracle Wallet Manager to install the trust points that are required to complete the chain. See Section 11.1.5.2.1.

Part V

Backup and Recovery

Backup and recovery refers to the various strategies and procedures involved in guarding against hardware failures and data loss, and reconstructing data should loss occur. This part describes how to back up and recover Oracle Application Server.

This part contains the following chapters:

15

Introduction to Backup and Recovery

This chapter provides information on getting started with Oracle Application Server backup and recovery.

It contains the following topics:

- Philosophy of Oracle Application Server Backup and Recovery
- Overview of the Backup Strategy
- Overview of Recovery Strategies
- What Is the OracleAS Recovery Manager?
- Assumptions and Restrictions
- Roadmap for Getting Started with Backup and Recovery

15.1 Philosophy of Oracle Application Server Backup and Recovery

This section introduces the philosophy for backing up and recovering your Oracle Application Server environment. An Oracle Application Server environment can consist of different components and configurations. To determine which components and configurations best meet your requirements, refer to the *Oracle Application Server Installation Guide* and *Oracle Application Server Concepts*.

A typical Oracle Application Server environment contains one or more middle-tier installations.

The installations in an Oracle Application Server environment are interdependent in that they contain configuration information, applications, and data that are kept synchronized. For example, when you perform a configuration change, you might update configuration files in the middle-tier installation; when you deploy an application, you might deploy it to all middle-tier installations.

It is, therefore, important to consider your entire Oracle Application Server environment when performing backup and recovery. You should back up your entire Oracle Application Server environment at once. Then, if a loss occurs, you can restore your entire environment to a consistent state.

The types of files for backup and recovery are:

- **Oracle software files**

  These are static files such as binaries and libraries. They reside in the middle-tier Oracle homes. They are created at installation time.

- **Configuration files**

These files contain configuration information and deployed applications. They reside in the middle-tier Oracle homes. They are created at installation or run time and are updated during the normal operation of your application server. The types configuration files include: Oracle HTTP Server, OC4J, and OPMN.

- **Oracle system files**

    These files may be in the `/var/opt/oracle` or `/etc` directory, and the `oraInventory` directory. They exist on each host in your Oracle Application Server environment. They usually reside outside of your Oracle Application Server installations, although the `oraInventory` directory may be in an Oracle home. They are created and updated by Oracle Universal Installer at installation time and contain information about your installations. On Windows, some registries are created by the installer.

The strategies and procedures in this book involve backing up and recovering these different types of files in a manner that maintains your Oracle Application Server environment in a consistent state.

> **Note:** Your Oracle Application Server environment contains additional files to those mentioned in this section, such as log files, and additional files you may deploy in the Oracle home, such as static HTML files and CGI scripts. You can add any of these files to the backup list.

15.2 Overview of the Backup Strategy

This section describes the backup strategy used in this book. It contains the following topics:

- Types of Backups
- Oracle Application Server Component Backup Input Files
- Plug-in Backup Input File
- Recommended Backup Strategy

15.2.1 Types of Backups

The Oracle Application Server backup strategy involves two types of backups:

- Image Backup
- Instance Backup

Image Backup

An image backup of an Oracle Application Server instance includes the Oracle home directory for that instance, the OraInventory directory, the oratab file, and Windows registries on that node and finally a cold instance backup of that Oracle Application Server instance. The Oracle home directory contains all the binary files, executables, initialization files, configuration files, log files, and so forth of the Oracle Application Server instance and of all components and deployed applications in that instance. The OraInventory directory contains the installation information for the instance.

Instance Backup

Contains the configuration information of Oracle Application Server components and deployed applications. The OracleAS Recovery Manager backs up all the local copies

of the configuration files specified for each configured middle-tier component in its backup input file.

15.2.2 Oracle Application Server Component Backup Input Files

Each Oracle Application Server component has a backup input file which contains a list of all the configuration files that should be backed up for that component. In a backup operation, if a component is installed and configured, the OracleAS Recovery Manager invokes the component's backup input file to determine what files to back up. A component backup input file has the file extension .inp and resides in the *Oracle_Home*/backup_restore/config directory. Table 15–1 lists the component backup input files that can reside in the directory.

Table 15–1   Oracle Application Server Component Backup Input Files

Component Name	Backup Input File
Oracle Enterprise Manager	config_em_files.inp
List of files to be excluded during backup	config_exclude_files.inp
Oracle Application Server installation information	config_install_files.inp
Oracle Enterprise Manager Log Loader	config_logloader_files.inp
Extra miscellaneous files to be backed up	config_misc_files.inp
Oracle Containers for J2EE applications	config_oc4j_files.inp
Oracle HTTP Server	config_ohs_files.inp
Oracle iASPT	config_iaspt_files.inp
Oracle Java Object Cache	config_javaobjcache_files.inp
Oracle Portal	config_portal_files.inp
Oracle Process Management and Notification Server	config_opmn_files.inp
Oracle WebCenter Framework metadata store	config_misc_files.inp

15.2.3 Plug-in Backup Input File

You create a plug-in backup input file as a way to add files to backups after installation. Each plug-in backup input file belongs to an Oracle Application Server component or an Oracle application and contains a list of additional files to be backed up.

The list of files specified by the plug-in backup input file must reside in the local Oracle home directory. The listed files are backed up when you run the backup_ config command. Since there may be interdependency between the data in these files and the data in all the other component configuration files, in the same Oracle home, Recovery Manager combines all the files into a single JAR archive file for restore operations. The first file in the plug-in file must be the key file and must be accessible by Recovery Manager. If Recovery Manager fails to locate and back up the first file, the entire backup configuration operation will terminate with a logged error message.

The format of entries in the plug-in backup input file is as follows:

To specify a particular file for backup:

```
${ORACLE_HOME}/directorypath/filename
```

To specify a directory:

```
${ORACLE_HOME}/directorypath
```

To use wildcards:

```
${ORACLE_HOME}/directorypath/*.conf
```

Wildcards cannot be used with the first entry in the plug-in backup input file list of files. The first file must be the key file and must be accessible by Recovery Manager.

After creating a plug-in backup input file, add the file to the *Oracle_Home*/backup_restore/plugin_config directory. The name of the plug-in backup input file must have the following format:

```
config_component_name_plugin.inp
```

Some examples are:

```
config_rules_plugin.inp
config_oc4j_plugin.inp
config_ohs1_plugin.inp
```

Enabling Plug-in Backup Input Files

Before Recovery Manager can backup the files specified in the plug-in backup input file, the plug-in backup input file must be enabled. The command `enable_component_inp` enables the input file. The following examples show the syntax for the command:

On UNIX:

```
bkp_restore.sh [-d -s -v] -m enable_component_inp -y "component_name[, component_name]…"
```

On Windows:

```
bkp_restore.bat [-d -s -v] -m enable_component_inp -y "component_name[, component_name]…"
```

For command and syntax information, see Section 16.4.2.

After enabling a plug-in input file, you must perform a new backup configuration operation (`backup_config`) before performing a restore configuration operation (`restore_config`).

15.2.4 Recommended Backup Strategy

This section outlines the recommended strategy for performing backups. Using this strategy ensures that you will be able to perform the recovery procedures in this book.

- **Perform a complete image backup.**

  Immediately after you install Oracle Application Server, you should perform a complete image backup for each node in your Oracle Application Server environment. This backup contains everything you need in order to restore each node to its initial state. It serves as a baseline for all subsequent run-time backups.

- **Perform instance backups on a regular basis.**

  After every administrative change, or, if this is not possible, on a regular basis, perform an instance backup of your Oracle Application Server environment. This enables you to restore your environment to a consistent state as of the time of your most recent configuration and applications backup. To avoid an inconsistent backup, do not make any configuration changes until backup completes for all Oracle Application Server instances.

> **See Also:**   Appendix E, "Examples of Administrative Changes" to learn more about administrative changes

- **After a major change, perform a new complete image backup.**

  If you make a major change to your Oracle Application Server environment, perform a new complete image backup. This backup will serve as the basis for subsequent online backups.

  Perform a new complete image backup after:

  - An operating system software upgrade

  - An Oracle Application Server software upgrade or patch application

  If you decide to back out an upgrade or patch, revert back to your last complete image backup. You can then apply any instance backups that occurred between the software upgrade or patch and the last complete image backup of your Oracle Application Server environment. Restoring an instance backup without restoring the last complete image backup might mix old configuration files with newly upgraded software that might not be compatible.

- **Perform instance backups on a regular basis.**

  After you establish a new complete image backup of your Oracle Application Server environment, continue to perform instance backups on a regular basis.

- **For extensive user customization and personalization of portlets, perform a portlet producer backup.**

  By doing so, the customization and personalization data, managed and stored by the portlet producer, is backed up. You must run the portlet producer customization and personalization backup utility on the node where the remote portlet producer runs.

15.3 Overview of Recovery Strategies

There are two types of Oracle Application Server recovery strategies used in this book:

- Recovery Strategies for Data Loss, Host Failure, or Media Failure (Critical)

- Recovery Strategies for Process Crashes or System Outages (Non-Critical)

Recovery Strategies for Data Loss, Host Failure, or Media Failure (Critical)

These 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

In all cases, these strategies involve making sure your state is consistent across all installations.

Recovery Strategies for Process Crashes or System Outages (Non-Critical)

These strategies involve restarting processes that have stopped or failed. They do not involve restoring data. They are included in this book for completeness.

15.4 What Is the OracleAS Recovery Manager?

The OracleAS Recovery Manager is an application that you can use to back up and recover configuration files in the middle tier.

The OracleAS Recovery Manager is installed by default whenever you install Oracle Application Server. It is installed in the *Oracle_Home*/backup_restore directory. See Section 16.2 for instructions on how to install OracleAS Recovery Manager manually.

15.5 Assumptions and Restrictions

The following assumptions and restrictions apply to the backup and recovery procedures in this book:

- OracleAS Recovery Manager is not backward compatible with previous releases of OracleAS Recovery Manager. Archives, created using previous versions of OracleAS Recovery Manager, are not recoverable by the current version.

- The following installation types are supported:
  - Oracle WebCenter Framework
  - Oracle WebCenter Framework with Oracle HTTP Server
  - Oracle HTTP Server

- The backup and recovery procedures do not support the Oracle Content Database installation type.

- If you are using OracleAS Cold Failover Cluster or Disaster Recovery, refer to the *Oracle Application Server High Availability Guide* for special considerations.

- For this release, you can run Recovery Manager from the command line only.

- On Windows, if you want to store backups on a remote file system, you must create a local mapped drive and specify it as the backup storage directory. For example, if Z:\ASbackups is the mapped drive for backups, then configuration files and repository backups should have Z:\ASbackups as their backup directory.

15.6 Roadmap for Getting Started with Backup and Recovery

This section provides a roadmap for getting started with Oracle Application Server backup and recovery.

1. **Configure OracleAS Recovery Manager.**

   Oracle recommends you configure OracleAS Recovery Manager and familiarize yourself with its features.

2. **Implement the backup strategy.**

   Chapter 17, "Backup Strategy and Procedures" outlines the Oracle-recommended backup strategy and backup procedures. Following this backup strategy ensures that you will be able to perform the recovery procedures in this book.

3. **Recover as necessary.**

   In the event of system failure or data loss, refer to Chapter 18, "Recovery Strategies and Procedures". It outlines different types of failures and describes the procedures you can follow to recover.

16

Oracle Application Server Recovery Manager

This chapter describes how to install, configure, and use the Oracle Application Server Recovery Manager.

It contains the following topics:

- How to Obtain OracleAS Recovery Manager
- How to Configure OracleAS Recovery Manager Manually
- Customizing OracleAS Recovery Manager for Your Configuration Files
- OracleAS Recovery Manager Usage Summary

16.1 How to Obtain OracleAS Recovery Manager

The OracleAS Recovery Manager is installed as part of an Oracle Application Server installation. It is located in the *Oracle_Home*/backup_restore directory. Table 16–1 lists the files that may reside in the backup_restore directory.

Table 16–1   OracleAS Recovery Manager Files

File[1]	Description
bkp_restore.sh	A shell script used to run the manager on UNIX.
bkp_restore.bat	A batch command file used to run the manager on Windows.
config/config.inp	The main configuration file that contains parameters for customizing OracleAS Recovery Manager for your environment. The orainst_loc_path field must be changed only if the instance is installed with the –invPtrLoc installer command-line option. It must be changed to reflect the nonstandard location of oraInst.loc.
config/config_*component*_files.inp	Component configuration files—each contains a list of configuration files for a particular component. These specify which files to back up when performing an instance backup. See Section 15.2.2 for a list component configuration files.

[1] Paths are relative to the root of the OracleAS Recovery Manager directory.

> **See Also:** *Oracle Application Server Installation Guide* for information about installing the Oracle Application Server.

16.2 How to Configure OracleAS Recovery Manager Manually

This section describes how to configure OracleAS Recovery Manager manually. You must follow these steps for each installation in your environment.

> **Note for Windows Users:** Do not use a rich text editor, such as WordPad, when editing files in the OracleAS Recovery Manager directory. It inserts a return character at the end of each line that may cause the manager to fail. Oracle recommends that you use a basic text formatter, such as Notepad, instead.

---

1. Prior to running the OracleAS Recovery Manager, set the ORACLE_HOME environment variable.

2. OracleAS Recovery Manager writes out log files and backup files, and you must specify the following directories to hold these. The default log file directory is *ORACLE_HOME*/backup_restore/logs. Edit config.inp to create the following directories:

   - **Log file directory**: This directory holds log files created by the manager. This directory should have several megabytes of space.

   - **Configuration file backup directory**: This directory holds configuration file backups. This directory should have several hundred megabytes of space.

   Recommendations for creating these directories are as follows:

   - Create your backup directories on a file system on a separate disk and, if possible, on a separate disk controller than where your Oracle Application Server Oracle home resides. This gives you the best chance of recovering data in the event of a hardware failure.

   - Make sure your backup directories are writable by the user that installed Oracle Application Server.

     For example, to create a log file directory and configuration file backup directory on /disk1:

     On UNIX:

     ```
     mkdir -p /disk1/backups/log_files
     mkdir -p /disk1/backups/config_files
     cd /disk1/backups
     chmod 755 log_files config_files
     chown OracleAS_user log_files config_files
     ```

     On Windows:

     ```
     mkdir C:\backups\log_files
     mkdir C:\backups\config_files
     ```

3. Edit config.inp and modify the parameters as described in Table 16–2.

Table 16–2   Parameters in config.inp

Parameter	Value
oracle_home (optional)	Do not insert a value for this. Using the command-line interface, set ORACLE_HOME in the shell environment first.
log_path (optional)	Specify the full path of the log file directory. If the full path is not specified, the default log directory *ORACLE_HOME*/backup_ restore/logs is automatically created when the -m configure command is executed. If a log_path is specified in the config.inp file, but the specified directory does not exist, OracleAS Recovery Manager automatically creates the specified log directory whether or not the -f (force) option is used in the -m configure command. However, the configuration file backup directory is not automatically created unless the -f option is specified.
config_files_list	Do not insert a value for this; leave it as config_files_ list=DO_NOT_SET.
	This parameter will be updated with the appropriate list of configuration files for your installation when you run bkp_ restore.pl -m configure.
config_backup_path (mandatory)	Specify the full path of the configuration file backup directory.
install_type	Do not insert a value for this; leave it as install_type=DO_ NOT_SET.
	This parameter is updated with the appropriate value for your installation when you run bkp_restore.pl -m configure.
orainst_loc_path (optional)	This parameter is used for UNIX platforms only. If the default path is overridden during installation, specify the full path of the directory where the oraInst.loc file exits. Otherwise, leave the parameter with the default value.
plugin_config_files_ list=DO_NOT_SET	Do not update this parameter. It is updated when you enable components containing plug-in backup input files.

Configure OracleAS Recovery Manager by running it with the -m configure option, for example:

■   For UNIX systems:

    ./bkp_restore.sh -m configure

■   For Windows systems:

    bkp_restore.bat -m configure

You are now ready to use OracleAS Recovery Manager.

16.3 Customizing OracleAS Recovery Manager for Your Configuration Files

As shipped, OracleAS Recovery Manager backs up all of the Oracle Application Server configuration files that are necessary to reconstruct an Oracle Application Server installation. You can customize OracleAS Recovery Manager to include any additional files that you would like to back up regularly, or to exclude any files you do not want to back up.

16.3.1 How OracleAS Recovery Manager Works When Backing Up Configuration Files

Before you customize OracleAS Recovery Manager, you should understand how it works. When you use the manager to back up your configuration files, it:

1. Opens `config.inp` (unless another environment file was specified with the `-e` option) and retrieves `config_files_list`.

2. Attempts to open each input file in `config_files_list` and exits with an error if it cannot open all of the files.

3. Examines the contents of `config_exclude_files.inp`. The manager will not attempt to back up the files listed in this file.

4. Walks through each file in `config_files_list` and examines the first entry in each file. This entry is the *key file*. The key file is used to determine if the component exists in this installation.

   - If the manager finds the key file, it knows the component is installed, and attempts to back up all of the entries in the file. It logs an error whenever it cannot find a key file. For all other files that the manager does not find, a warning is issued and the backup continues.

   - If the key file does not exist, the manager does not attempt to back up any entries in the component input file. It logs an error to the log file and skips to the next component input file.

5. The configuration files are stored in a jar file located in the directory specified by the `config_backup_path` parameter in the config.inp file:

   `config_bkp_2006-05-10_18-33-10.jar`

16.3.2 How to Customize OracleAS Recovery Manager

Because OracleAS Recovery Manager knows how to determine which configuration files exist in your installation, it is not necessary to customize OracleAS Recovery Manager. However, you may want to customize the manager by:

- Adding Files to a Backup

  You may want to add your own local configuration files or any other files you would like to back up regularly, such as log files.

- Excluding Files from a Backup

  You may want to exclude files from being backed up.

Adding Files to a Backup

To add files, such as Oracle Application Server component specific log files, to a backup, add entries to the `config_misc_files.inp` file as follows:

- To specify a particular file:

  `${ORACLE_HOME}/directorypath/file`

- To specify an entire directory:

  `${ORACLE_HOME}/directorypath/`

- To use wildcards:

  `${ORACLE_HOME}/directorypath/*.html`

You can add as many entries as you like. The `config_misc_files.inp` file is always included in the `config_files_list` parameter in `config.inp`, so there is no need to edit `config.inp`.

In some cases, OracleAS Recovery Manager might not be aware of additional files stored outside a typical directory structure. For example, in following cases you must edit `config_misc_files.inp` to ensure proper backup of the additional files:

- Virtual or nondefault paths defined in Oracle HTTP Server configuration files such as `httpd.conf` or `moddav.conf`. If you change these web server configuration files to point to other files or directories, you should consider making the new paths a part of the run-time backup.

- An application deployed to a OC4J container that uses files located outside the container directory. OracleAS Recovery Manager automatically backs up all the files located in the container directory. If your application uses any additional directories, you should consider them as part of configuration backups.

- Java Messaging Service (JMS) with the file-based persistence. The JMS run-time data (messages) are stored in physical files and should be a part of the backup process.

Note that you do not need to specify a key file in `config_misc_files.inp`.

Excluding Files from a Backup

You can exclude files from a backup in either of the following ways:

- You can simply remove the file entry from its `config_component.inp` file.

- If you have a situation where a `config_component.inp` file specifies an entire directory to back up, and you would like to exclude a specific file from that directory, you can add an entry for that file to `config_exclude_files.inp`. The manager will back up the entire directory except for the file you specify. You cannot specify directories or use wildcards in `config_exclude_files.inp`. Only single file entries are allowed.

  Note that you do not need to specify a key file in `config_exclude_files.inp`.

16.4 OracleAS Recovery Manager Usage Summary

This section summarizes usage for OracleAS Recovery Manager.

It contains the following topics:

- Prerequisites for Running OracleAS Recovery Manager
- Syntax
- Usage Examples
- Purging Backups and Moving Them to Tertiary Storage

16.4.1 Prerequisites for Running OracleAS Recovery Manager

Before running OracleAS Recovery Manager:

- Log in as the user that installed Oracle Application Server.
- Make sure the `ORACLE_HOME` environment variable is set.

16.4.2 Syntax

The syntax for OracleAS Recovery Manager is:

On UNIX:

```
bkp_restore.sh [-defsv] -m mode [args]
```

On Windows:

```
bkp_restore.bat [-defsv] -m mode [args]
```

It accepts the following options:

-d Print a trace without executing.
-e Specify an environment file (default is `config.inp`).
-f Force log file and configuration file directories to be created if they are required by the current command and do not exist.
-n Suppress prompts so the manager can run in batch mode.
-o Loss of Host Automation (LOHA) operation
-s Run in silent mode.
-v Run in verbose mode.
-y Enable components associated with plug-in backup input files.

Use the -m option to specify which mode to run. Some modes take arguments. Table 16–3 describes OracleAS Recovery Manager modes and their arguments. All modes and arguments are case-sensitive.

A limitation of at least one minute between `bkp_restore` operations is enforced. If there is a backup jar present with a timestamp with less than minute difference from the backup operation currently running, the current backup operation will fail.

Table 16–3   OracleAS Recovery Manager Modes and Arguments

Mode and Arguments	Description
`backup_config`	Performs a full configuration backup. The command performs the following operations:
	■ Opens `config.inp` (or the alternate file specified with the -e option) and retrieves `config_files_list`, `config_backup_path`, and `log_path`.
	■ Attempts to open each file in `config_files_list`. Exits with an error if it cannot open all of the files.
	■ For each file in `config_files_list`, checks if the first entry (the key file) exists. If the key file does not exist, it is treated as a fatal error. Otherwise, backs up all files in the list. If any other files do not exist, logs an error and continues.
	■ Excludes files listed in `config_exclude_files.inp`.
	■ When finished, stores the backup in `config_backup_path/config_bkp_timestamp`.
	■ If any errors are encountered, creates a log file in `log_path/config_bkp_timestamp`.
`backup_config_incr`	Performs an incremental configuration file backup.
	Works the same as `backup_config`, except it backs up all configuration files that have changed since the last full or incremental configuration file backup.

Table 16–3   (Cont.) OracleAS Recovery Manager Modes and Arguments

Mode and Arguments	Description
`backup_instance_cold`	Performs a complete cold backup of the Oracle Application Server instance. The command performs the following operations:
	■ Stops all OPMN managed processes.
	■ Starts the OPMN administrative process.
	■ Checks all of the OPMN managed processes to ensure that the processes are stopped. If not, tries to stop them one more time. If the processes still cannot be stopped, issues a fatal error.
	■ Performs a full configuration backup.
	■ Starts all OPMN managed processes.
	■ Checks to ensure that all OPMN processes are running. If not, issues a warning message.
`backup_instance_cold_ incr`	Performs an incremental cold backup of the Oracle Application Server instance. The command performs the following operations:
	■ Stops all OPMN managed processes.
	■ Starts the OPMN administrative process.
	■ Checks all of the OPMN managed processes to ensure that the processes are stopped. If not, tries to stop them one more time. If the processes still cannot be stopped, issues a fatal error.
	■ Performs incremental configuration backup.
	■ Starts all OPMN managed processes.
	■ Checks to ensure that all OPMN processes are running. If not, issues a warning message.
`backup_instance_online`	Performs an online backup of the Oracle Application Server instance.
`backup_instance_online_ incr -1 level number`	Performs an incremental online backup of the Oracle Application Server instance.
`configure`	Configures the manager. The command updates `config_files_list` and `install_type` in `config.inp` with the appropriate information for your installation.
`help`	Prints a usage message.
`list_backups`	Lists the configuration backups taken for the instance.
`list_instance_backups`	Lists instance-level backups taken for the instance.
`list_changed_config`	Lists any configuration files that have changed since the last full or incremental backup. This command checks the modification date of each file; it does not check the actual contents of the file. It writes the list of files to a log file and prints the name of the log file. Deleted files or deleted directories are not listed in `list_changed_config`. Only modified files or directories containing modified files are listed.
`node_backup  -o image_ backup -P directory for the image archive`	Creates an image archive of the original host. The image includes the original Oracle home, oratab, central inventory and so forth, depending on the installation. On UNIX, this operation must be run as `root`.
`node_backup -o prepare`	Prepares the node for backup. Preparation includes discovering the operating system type, host name/IP address, user/group ID, install type, the location of the central inventory, Oracle home locations if there are multiple of them, Windows registry, Windows service database scanning to find all services created for Oracle homes. The information is placed in a file to be used in node restoration.
	This mode also creates a config backup.

Table 16–3   (Cont.) OracleAS Recovery Manager Modes and Arguments

Mode and Arguments	Description
`node_restore -o inst_reconfigure -t config_bkp_timestamp`	Reconfigures the instance on the new host including IP changes, config backup, restore and so forth, depending upon the installation type.
`node_restore -o inst_register`	Registers the instance with the oratab and the central inventory. It also sets up the daemon start and stop script and so forth by running `root.sh`. On Windows, Windows services are created.
	It must be run as `root` on UNIX systems.
`node_restore -o sys_init`	Restores Oracle Universal Installer related metadata such as oratab (UNIX), Windows registries (Windows) and central inventory. It should be run once only on the new host.
	It must be run as `root` on UNIX systems.

`restore_config`
 	
`[-t config_bkp_`
`timestamp]`
 	
`[-n]`
	Restores configuration files. The command performs the following operations:
	
	■ Opens `config.inp` (or the alternate file specified with the `-e` option) and retrieves `config_backup_path` and `log_path`.
	
	■ If the `-t` option is supplied and it is the timestamp from a full backup, restores that full backup.
	
	■ If the `-t` option is supplied and it is the timestamp from an incremental backup, restores the full backup and all incremental backups, up to and including the specified incremental backup.
	
	■ If the `-t` option is not supplied, displays a list of configuration file backups in `config_backup_path` and exits. You can then rerun the command and supply one of these files with the `-t` option.
	
	■ Restores all files from the configuration file backup to the Oracle home, preserving owner, group, permissions, and timestamp.
	
	■ If any errors are encountered, creates a log file in `log_path`/config_rst_*timestamp*.
	
	The `-n` option suppresses prompts so you can use the manager in batch mode.
	
	For the process prerequisites, refer to the `backup_config` option.
	
	Do not run `restore_config` on multiple nodes in a J2EE cluster in parallel. Doing so will cause `restore_config` failures. Run `restore_config` on one node at a time.

`restore_instance -t` `timestamp`	Restores an instance of Oracle Application Server. If the timestamp argument is not specified, then a list of backup timestamps displays. The command performs the following operations: ■ Stops all OPMN managed processes. ■ Checks to verify that the OPMN processes have stopped. If OPMN processes cannot be stopped (maybe an `opmn.xml` file is missing), a file system restore is performed. Then, tries to stop the OPMN processes again. If the OPMN processes still cannot be stopped, issues a fatal error. ■ Starts the OPMN administration process. ■ Performs configuration restore. ■ Starts all OPMN managed processes. ■ Checks to ensure that all OPMN managed processes are started. If not, issues a warning message.

16.4.3 Usage Examples

This section contains usage examples for OracleAS Recovery Manager. The UNIX command is listed first and then the Windows command.

- Configure the manager using the default `config.inp` file:

```
bkp_restore.sh -m configure
bkp_restore.bat -m configure
```

- Configure the manager using a configuration file called `myconfig.inp`:

```
bkp_restore.sh -m configure -e myconfig.inp
bkp_restore.bat -m configure -e myconfig.inp
```

- Perform an cold backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_cold
bkp_restore.bat -m backup_instance_cold
```

- Perform an incremental cold backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_cold_incr
bkp_restore.bat -m backup_instance_cold_incr
```

- Perform an online backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_online
bkp_restore.bat -m backup_instance_online
```

- Perform an online incremental backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_online_incr
bkp_restore.bat -m backup_instance_online_incr
```

- Restore an Oracle Application Server instance to its state at a particular time:

```
bkp_restore.sh -m restore_instance -t 2006-09-21_06-12-45
bkp_restore.bat -m restore_instance -t 2006-09-21_06-12-45
```

- Node backup preparation using Loss of Host Automation (LOHA):

```
bkp_restore.sh -m node_backup -o prepare
bkp_restore.bat -m node_backup -o prepare
```

- Create an image backup of the original host using LOHA:

```
bkp_restore.sh -m node_backup -o image_backup -P directory for image archive
bkp_restore.bat -m node_backup -o image_backup -P directory for image archive
```

- Restore Oracle Universal Installer related metadata on the new host using LOHA:

```
bkp_restore.sh -m node_restore -o sys_init
bkp_restore.bat -m node_restore -o sys_init
```

- Register the instance on the new host using LOHA:

```
bkp_restore.sh -m node_restore -o inst_register
bkp_restore.bat -m node_restore -o inst_register
```

- Configure the instance on the new host using LOHA:

```
bkp_restore.sh -m node_restore -o inst_reconfigure -t config_bkp_timestamp
bkp_restore.bat -m node_restore -o inst_reconfigure -t config_bkp_timestamp
```

16.4.4 Purging Backups and Moving Them to Tertiary Storage

OracleAS Recovery Manager saves records of successful backups in a catalog file (`data/catalog.txt`) in the `backup_restore` directory. Each backup is identified

by a timestamp, which is also embedded in the file names of jar files saved in the configuration file backup directory in the case of a instance or configuration-only backup. If you delete all the `.jar` files corresponding to a timestamp or move them somewhere else, for example offline storage, although the catalog still contains a record of the timestamp, you will not see this record when you run `-m list_ backups`, nor will you be able to restore using this timestamp as the `-t` value. This is the expected behavior.

17

Backup Strategy and Procedures

This chapter describes the Oracle Application Server backup strategy and procedures.

It contains the following topics:

- Recommended Backup Strategy
- Backup Procedures
- Recovering a Loss of Host Automatically

17.1 Recommended Backup Strategy

This section describes the recommended backup strategy for Oracle Application Server. Using this strategy ensures that you can perform the recovery procedures described in this book.

The backup strategy is as follows:

- Task 1: Perform a Complete Cold Backup of Your Oracle Application Server Environment
- Task 2: Perform Instance Backups on a Regular Basis
- Task 3: Perform a New Complete Environment Backup After a Major Change
- Task 4: Perform Instance Backups on a Regular Basis (Return to Task 2)
- Task 5: Perform Portlet Producer Backup

The flow chart in Figure 17–1 provides an overview of how to decide which type of backup is appropriate for a given circumstance.

Figure 17–1   Deciding the Type of Backup Needed
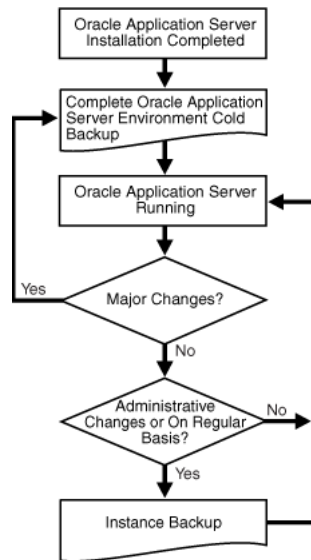
Task 1: Perform a Complete Cold Backup of Your Oracle Application Server Environment

The first backup you perform should be an image backup, which includes all of the files in your environment. You should also create a record of your environment.

1. Perform a complete Oracle Application Server environment backup.

This will serve as the baseline for all subsequent instance backups.

Refer to Section 17.2.3.

2. Create a record of your Oracle Application Server environment.

In the event you need to reconstruct your environment, you can refer to this record.

Refer to Section 17.2.1.

Task 2: Perform Instance Backups on a Regular Basis

After every administrative change, or, if this is not possible, on a regular basis, perform an instance backup of your Oracle Application Server environment.

> **See Also:**   Appendix E, "Examples of Administrative Changes" to learn more about administrative changes

Refer to Section 17.2.2.

Task 3: Perform a New Complete Environment Backup After a Major Change

If you make a major change to your Oracle Application Server environment, you must perform a new image backup of your Oracle Application Server environment. This backup serves as the basis for subsequent instance backups. You should also update the record of your environment with any new configuration information.

Perform a new image backup after:

■   An operating system software upgrade

■   An Oracle Application Server software upgrade or patch application

To do so:

1. Update the record of your Oracle Application Server environment.

   Refer to Section 17.2.1.

2. Perform a complete Oracle Application Server environment backup.

   Refer to Section 17.2.3.

Task 4: Perform Instance Backups on a Regular Basis (Return to Task 2)

After you establish a new complete Oracle Application Server environment backup, return to Task 2 and continue to perform instance backups on a regular basis.

Task 5: Perform Portlet Producer Backup

If your application contains remote portlet producers, you must back up the portlet producer customization and personalization data.

Additional Tips:

- Create a backup of the JRE/JDK on your system. This is not an Oracle product, but it is utilized by Oracle Application Server and, if accidentally lost or corrupted, would need to be restored in order for Oracle Application Server to function. This issue only applies to HP-UX, HP Tru64, and IBM AIX systems.

- Ensure that your backups are valid by routinely verifying that they can be restored.

17.2 Backup Procedures

This section describes the backup procedures in detail. To maintain configuration data consistency, you should back up each of your Oracle Application Server instances at the same time. While backing up one Oracle Application Server instance, ensure that no configuration changes are made in any of the other instances.

This section contains the following topics:

- Creating a Record of Your Oracle Application Server Configuration

- Performing an Oracle Application Server Instance Backup from the Command Line

- Performing a Complete Oracle Application Server Environment Backup

- Performing a Portlet Producer Backup

17.2.1 Creating a Record of Your Oracle Application Server Configuration

In the event you need to restore and recover your Oracle Application Server 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 Application Server environment on a new disk or host.

You should maintain an up-to-date record of your Oracle Application Server 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 Application Server environment.

Your Oracle Application Server hardware and software configuration record should include:

- The following information for each host in your environment:
  - Hostname
  - Virtual hostname (if any)
  - Domain name
  - IP address
  - Hardware platform
  - Operating system release level and patch information
- The following information for each Oracle Application Server installation in your environment:
  - 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 `ORACLE_HOME`
  - Amount of disk space used by the installation
  - Port numbers used by the installation

> **Note:** Use `opmnctl status -l` to determine the ports in use.

17.2.2 Performing an Oracle Application Server Instance Backup from the Command Line

This section describes how to perform various Oracle Application Server instance backups from the command line. An instance-level backup backs up all the required components in an application server instance: configuration files, repositories for the middle tier.

Once you have performed a complete Oracle Application Server environment backup, you should perform subsequent instance-level backups after every administrative change, or, if this is not possible, on a regular basis.

Performing a Cold Backup of an Oracle Application Server Instance

Use the following command to perform a cold backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_cold
bkp_restore.bat -m backup_instance_cold
```

Performing an Incremental Cold Backup of an Oracle Application Server Instance

Use the following command to perform an incremental cold backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_cold_incr
bkp_restore.bat -m backup_instance_cold_incr
```

Performing an Online Backup of an Oracle Application Server Instance

Use the following command to perform an online backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_online
bkp_restore.bat -m backup_instance_online
```

Performing an Incremental Online Backup of an Oracle Application Server Instance

Use the following command to perform an incremental online backup of an Oracle Application Server instance:

```
bkp_restore.sh -m backup_instance_online_incr -l level
bkp_restore.bat -m backup_instance_online_incr -l level
```

17.2.3 Performing a Complete Oracle Application Server Environment Backup

This section describes how to perform a complete Oracle Application Server environment backup. You should back up the node after installation or after an upgrade. Perform the following tasks for each instance on the host:

Configuration Backup of the Node

Run the following command to create a backup of the node configuration:

On UNIX:

```
bkp_restore.sh -m configure
```

On Windows:

```
bkp_restore.bat -m configure
```

Node Backup Preparation

Run the following command to prepare a node for backup:

On UNIX:

```
bkp_restore.sh -m node_backup -o prepare
```

On Windows:

```
bkp_restore.bat -m node_backup -o prepare
```

Creating an Image Backup of the Instance

This task creates an archive of an instance that includes the Oracle home, oratab, central inventory, Windows registries and so forth. On UNIX, the command must be run from root. Run the following command to create an image backup of the instance:

On UNIX:

```
bkp_restore.sh -m node_backup  -o image_backup -P archive path
```

On Windows:

```
bkp_restore.bat -m node_backup  -o image_backup -P archive path
```

After the command completes, the backup is placed in the directory specified in *archive path*.

17.2.4 Performing a Portlet Producer Backup

To completely back up and recover an application containing remote portlet producers, you must back up and recover two additional items:

- The portlet producer Web application itself. The producer's application is backed up and recovered as part of the overall Oracle Application Server backup and recovery. If the portlet producer runs on a remote Oracle Application Server installation, that application server must also be backed up.

- The producer's preference store, which contains portlet personalization and customization data. You must use a preference store migration utility to back up the preference store.

Using the Predeployment tool, you can use the `-backup` option to specify the location of the Oracle Metadata Services (OMS) repository when configuring OMS. This allows the Recovery Manager to backup the OMS repository by adding the location to the `config_misc_files.inp` file.

> **Note:** Portlet preferences can be stored either in a relational database or in a file system. For more information on configuring the portlet preference store, refer to *Oracle WebCenter Framework Developer's Guide*.

> **Note:** For information on disaster backup and recovery, refer to the Disaster Recovery section of the *Oracle Application Server High Availability Guide*.

To back up your portlet producer preference store, you can use the JPS and PDK-Java preference store migration utilities.

- [JPS Preference Store Backup](#)

- [PDK-Java Preference Store Backup](#)

17.2.4.1 JPS Preference Store Backup

To back up your JPS preference store, perform the following steps:

1. Stop the OC4J instance on which the portlet container runs.

2. Run the migration tool to back up data from the source preference store to the destination (backup) store. For example:

```
java -classpath
wsrp-container.jar:cache.jar:saaj-api.jar:orasaaj.jar:ojdbc14.jar
oracle.portlet.server.containerimpl.PersistenceMigrationTool
-sourceType db
-destType file
-sourceDatabase portaldb.mycompany.com:1521:orcl
-sourceUsername p1
-sourcePassword p1
-destPath /tmp/portletbkp
```

3. Start the OC4J instance on which the portlet container runs.

PersistenceMigrationTool Syntax

The syntax of the `PersistenceMigrationTool` is:

```
java oracle.webdb.wsrp.server.PersistenceMigrationTool
-sourceType file | db
-destType file | db
{-sourcePath dir |
 -sourceUsername username -sourcePassword password -sourceDatabase db}
```

```
{-destPath dir | destUsername username -destPassword password -destDatabase db}
[-debug]
```

where:

`sourceType` indicates whether the source store is in a file or database. You may have source and destination stores of the same type. Hence, you can migrate from one database to another or one file system to another.

`destType` indicates whether the destination store is in a file or database. You may have source and destination stores of the same type. Hence, you can migrate from one database to another or from one file system to another.

`sourcePath` is the location of a file-based preference store. This argument is required when `sourceType` is `file`.

`sourceUsername` is the database user name for a preference store database. This argument is required when `sourceType` is `db`.

`sourcePassword` is the database password for a preference store database. This argument is required when `sourceType` is `db`.

`sourceDatabase` is the name of a preference store database. This argument is required when `sourceType` is `db`.

`destPath` is the location of a file-based preference store. This argument is required when `destType` is `file`.

`destUsername` is the database user name for a preference store database. This argument is required when `destType` is `db`.

`destPassword` is the database password for a preference store database. This argument is required when `destType` is `db`.

`destDatabase` is the name of a preference store database. This argument is required when `destType` is `db`.

`debug` turns on full logging through standard output to allow users to diagnose issues that arise when the tool runs.

> **Note:** You can also obtain this syntax from the command line by entering the following command:
>
> ```
> java -classpath
> ORACLE_HOME/j2ee/OC4J_WebCenter/shared-lib/oracle.wsrp/1.0/
>     wsrp-container.jar:
> ORACLE_HOME/javacache/lib/cache.jar:
> ORACLE_HOME/j2ee/OC4J_WebCenter/webservices/lib/saaj-api.jar:
> ORACLE_HOME/j2ee/OC4J_WebCenter/webservices/lib/orasaaj.jar:
> ORACLE_HOME/j2ee/OC4J_WebCenter/jdbc/lib/ojdbc14.jar
> oracle.portlet.server.containerimpl.PersistenceMigrationTool
> ```

Preference Store Specification Example

To discover the type of a WSRP producer's preference store, review its `web.xml` file (Example 17–1).

Example 17–1   The persistentStore and fileStoreRoot Variables in the web.xml File

```
<env-entry>
    <env-entry-name>oracle/portal/wsrp/server/persistentStore</env-entry-name>
    <env-entry-type>java.lang.String</env-entry-type>
```

```
        <env-entry-value>File</env-entry-value>
</env-entry>
<env-entry>
    <env-entry-name>oracle/portal/wsrp/server/fileStoreRoot</env-entry-name>
    <env-entry-type>java.lang.String</env-entry/type>
    <env-entry-value>portletdata</env-entry-value>
</env-entry>
```

For more information about the JPS portlet migration utility, refer to *Oracle WebCenter Framework Developer's Guide*.

17.2.4.2 PDK-Java Preference Store Backup

To back up your PDK-Java preference store, perform the following steps:

1. Stop the OC4J instance on which the portlet container runs.

2. Run the migration tool to back up data from the source preference store to the destination (backup) store. For example:

```
java -classpath $ORACLE_HOME/portal/jlib/pdkjava.jar
 oracle.portal.provider.v2.preference.MigrationTool
 -mode dbtofile
 -remap locale
 -countries AR,MX
 -pref1UseHashing true
 -pref1User portlet_prefs
 -pref1Password portlet_prefs
 -pref1URL jdbc:oracle:thin:@myserver.mydomain.com:1521:mysid
 -pref2RootDirectory /tmp/portletbkp
```

3. Start the OC4J instance on which the portlet container runs.

Migration Tool Syntax

The syntax of the migration utility is:

```
java -classpath $ORACLE_HOME/portal/jlib/pdkjava.jar
 oracle.portal.provider.v2.preference.MigrationTool
  -mode [file | db | filetodb | dbtofile | dbtodb]
  [-remap language | locale]
  [-countries iso_country_code]
  [-pref1UseHashing true | false]
  {-pref1RootDirectory directory |
   -pref1User username -pref1Password password -pref1URL url}
  [-pref1UseHashing true | false]
  {-pref2RootDirectory directory |
   -pref2User username -pref2Password password -pref2URL url}
  [-upfixwpi filename]
```

where:

-mode is the mode in which you want to run the Preference Store Migration and Upgrade Utility.

- file or db indicates that you want to run in upgrade mode.

- filetodb, dbtofile, or dbtodb indicates that you want to run in migration mode.

-remap is the localePersonalizationLevel (language or locale). Note that you only need to use this option if you want to change localePersonalizationLevel as part of your upgrade or migration.

-`countries` specifies a prioritized list of ISO country codes, indicating your order of preference in case of a collision between remapped preferences for different countries. -`countries` is only meaningful if you also specified the -`remap` option.

-`pref1UseHashing` indicates whether or not you want to employ hashing on the source for this operation.

-`pref1RootDirectory` is the path of a source file system, for example, `j2ee/home/applications/jpdk/jpdk/WEB-INF/providers/sample`.

-`pref1User` is the user name for a source database.

-`pref1Password` is the password for a source database.

-`pref1URL` is the URL to a source database, for example, `jdbc:oracle:thin:@myserver.mydomain.com:1521:mysid`.

-`pref2UseHashing` indicates whether or not you want to employ hashing on the destination for this operation.

-`pref2RootDirectory` is the path of a destination file system, for example, `j2ee/home/applications/jpdk/jpdk/WEB-INF/providers/sample`.

-`pref2User` is the user name for a destination database.

-`pref2Password` is the password for a destination database.

-`pref2URL` is the URL to a destination database, for example, `jdbc:oracle:thin:@myserver.mydomain.com:1521:mysid`.

-`upfixwpi` indicates a log file for the operation.

> **Note:** You can also obtain this syntax from the command line by entering the following command:
>
> ```
> java -classpath C:\JDEV_HOME\adfp\lib\pdkjava.jar;
>   C:\jdev_10132_wcs_4007\adfp\lib\ptlshare.jar
>   oracle.portal.provider.v2.preference.MigrationTool
> ```

Preference Store Specification Example

To discover the type of a PDK-Java producer's preference store, review its `provider.xml` file (Example 17–2).

Example 17–2

```
<provider class="oracle.portal.provider.v2.DefaultProviderDefinition">
  <localePersonalizationLevel>none</localePersonalizationLevel>
  <session>true</session>
  <defaultLocale>en</defaultLocale>
  <preferenceStore
    class="oracle.portal.provider.v2.preference.FilePreferenceStore">
    <name>prefStore1</name>
  </preferenceStore>
  <portlet
   class="oracle.portal.sample.v2.devguide.prefstore.GuestBookPortletDefinition">
   <id>1</id>
   <name>GuestBook</name>
   <title>Guest Book Portlet</title>
   <shortTitle>Guest Book</shortTitle>
   <description>Demonstration of using a Preference Store to drive
     portlet content</description>
   <timeout>100</timeout>
```

```
        <timeoutMessage>Guest Book Portlet timed out</timeoutMessage>
        <renderer class="oracle.portal.provider.v2.render.RenderManager">
         <showPage>/htdocs/prefstore/guest_book.jsp</showPage>
         <editPage>/htdocs/prefstore/store_comment.jsp</editPage>
        </renderer>
      </portlet>
</provider>
```

For more information on the PDK-Java migration and upgrade utility, refer to *Oracle WebCenter Framework Developer's Guide*.

17.3 Recovering a Loss of Host Automatically

OracleAS Recovery Manager provides an automated procedure to take a full backup of the instances on one host and restore them to a new host after losing the original operating environment.

Loss of Host Automation (LOHA) automates the tasks necessary for the Oracle Application Server administrator to migrate Oracle Application Server instances from one host to another. The new host can be a different host running the same operating system or the same host after system re-imaging. LOHA provides a solution for a loss of host when you want to restore the original instances to a new environment without having to reinstall the instances and preserve the application data.

LOHA supports all middle-tier installations, and the new host's name can be the same or different from the original host. For different host names, some manual work is required.

LOHA can move all the Oracle Application Server instances from one host to a new host if the new host does not have any other Oracle Application Server instances already running. You can restore a subset of the instances to the new host if the subset does not have any dependencies on the instances remaining on the old host. You cannot restore instances from multiple hosts to a single host.

LOHA can also be used to recover a corrupted instance on a host without affecting other instances on the same host.

This section contains the following topics:

- Preparing to Use Loss of Host Automation

- Enabling Loss of Host Automation

- Restoring a Node on a New Host

- Recovering an Instance on the Same Host

17.3.1 Preparing to Use Loss of Host Automation

The Loss of Host Automation service is installed as part of OracleAS Recovery Manager. It is installed into the following directory:

On UNIX:

ORACLE_HOME/backup_restore/loha

On Windows:

ORACLE_HOME\backup_restore\loha

To use the Loss of Host Automation service, you must configure OracleAS Recovery Manager as described in Chapter 16, "Oracle Application Server Recovery Manager".

The Loss of Host service has the following prerequisites:

- The new host must have the same version of operating system and the same level of patches as required by Oracle Application Server.

- In the `config.inp` file, the `orainst_loc_path` field must be changed only if the instance is installed with the `-invPtrLoc` installer command line option. It must be changed to reflect the nonstandard location of `oraInst.loc`.

- For Windows platforms, Windows Support Files (WSF) must be installed. You can obtain WSF from the Oracle Application Server installation CDROM. For instructions on installing Windows system files, see *Oracle Application Server Installation Guide*.

- For Windows platforms, the Microsoft service utility `sc.exe` must be installed on both the original host and the new host. According to Microsoft, it is part of the NT ResourceKit. For Windows XP, the utility is part of the installation. For Windows 2000 platforms, it must be installed. Ensure that it is in the execution path.

- For UNIX platforms, ensure that the ORACLE_HOME environment variable does not have a trailing forward slash (/).

- On the new host, `jar` (Windows) or `tar` (UNIX) must be available to unpack the node archive. If your system has its own tar program, use it instead of GNU tar.

- The user must have administrative privileges on the system such that system or root-level tasks can be performed.

- There should not be any other Oracle products installed on the new host. For example, if there are some Oracle Application Server instances on this new host, they must be shutdown and uninstalled cleanly.

- The user/group ID on the new host must match that on the original host.

- Check port usage on the new host. Make sure there are no processes using the same ports as any of the Oracle Application Server instances you are restoring. If any processes are using the same ports, reconfigure the processes to use different ports before restoring any Oracle Application Server instance.

- After completing the restore, the same mount point and full path as the original middle-tier Oracle home are preserved. Ensure that the Oracle home parent directory is on a file system with enough space to hold the middle-tier installation, and that the directory is owned by the same user and group as on the original host.

17.3.2 Enabling Loss of Host Automation

The following tasks must be performed, for each instance on the original host, to enable the Loss of Host Automation service:

Configuration Backup of the Node
You should back up the node after installation or after an upgrade. Run the following command to create a backup of the node configuration:

On UNIX:

```
bkp_restore.sh -m configure
```

On Windows:

```
bkp_restore.bat -m configure
```

Node Backup Preparation

During node backup preparation, the Loss of Host Automation service determines the following information about the current host:

- Operating system

- Host name

- IP address

- User/group ID

- Install type

- Central inventory location

- Oracle home locations

- Windows registry and all Windows services created for all Oracle homes

The service also creates an instance backup with this operation.

Run the following command to prepare a node for backup:

On UNIX:

```
bkp_restore.sh -m node_backup -o prepare
```

On Windows:

```
bkp_restore.bat -m node_backup -o prepare
```

Creating an Image Backup of the Original Host

This task creates an archive of an instance that includes the original Oracle home, oratab, central inventory, Windows registries and so forth. On UNIX, the command must be run from `root`. Run the following command to create an image backup of the original instance:

On UNIX:

```
bkp_restore.sh -m node_backup  -o image_backup -P archive path
```

On Windows:

```
bkp_restore.bat -m node_backup  -o image_backup -P archive path
```

After the command completes, the backup is placed in the directory specified in *archive path*.

17.3.3 Restoring a Node on a New Host

The commands in this section restore a node on a new host after a loss of host. Before performing the following steps, ensure that all the prerequisites in Section 17.3.1 are fulfilled.

The following commands must be run in order.

1. Unpack the backup archive of the old node:

   On UNIX, login as root:

   ```
   cd /
   tar -xvpf archive_name
   ```

   On Windows:

```
jar -xvf archive_name
```

2. The following command restores Oracle Universal Installer related metadata such as oratab (UNIX), Windows registries, and central inventory on the new host. If multiple instances are to be restored, this operation should be performed only for the first instance. The command must be run as `root` on UNIX.

   On UNIX:

   ```
   bkp_restore.sh -m node_restore -o sys_init
   ```

   On Windows:

   ```
   bkp_restore.bat -m node_restore -o sys_init
   ```

3. The following command registers the instance with oratab and the central inventory; it also sets up daemon start/stop script by running `root.sh` on UNIX, or, on Windows, it creates Windows services. The command must be run as `root` on UNIX.

   On UNIX:

   ```
   bkp_restore.sh -m node_restore -o inst_register
   ```

   On Windows:

   ```
   bkp_restore.bat -m node_restore -o inst_register
   ```

4. This command reconfigures the instance on the new host. This includes IP changing, config backup restore and so forth, depending on the install type. Prior to running the command, run `opmnctl shutdown` to ensure that opmn and Enterprise Manager processes are not using ports required by the reconfigure process. For Infrastructure and metadata repository installations on Windows, the flashback_recovery_area must be manually created before running the command. The command must be run as the owner of the instance. The path to the instance backups must be valid.

   On UNIX:

   ```
   bkp_restore.sh -m node_restore -o inst_reconfigure -t config_bkp_timestamp
   ```

   On Windows:

   ```
   bkp_restore.bat -m node_restore -o inst_reconfigure -t config_bkp_timestamp
   ```

   Without a timestamp argument, this command shows all the available instance backups. For a successful completion of this operation, ensure that all the other required services are started if they do not belong to this instance.

   LOHA will not detect port conflicts on the new host. It is recommended that you do not run other applications using the same TCP ports that are to be used by the restored instance. Any port conflict will cause this operation to fail.

5. If an `Oracle_Home/backup_restore/config/config_misc_files.inp_<time_stamp>` file exists and is more recent than the existing `config_misc_files.inp` file, overwrite the `config_misc_files.inp` file with the most recent `config_misc_files.inp_<time_stamp>` file.

   For each component plug-in file in the `Oracle_Home/backup_restore/plugin_config` directory, check to see if there is a timestamped version that is more recent than the `config_component_name_plugin.inp`

file. If there is a more recent version, overwrite the `config_component_name_plugin.inp` file with the most recent timestamped version.

17.3.4 Recovering an Instance on the Same Host

When an instance of Oracle Application Server requires an image restore to correct a problem, you can use LOHA to recover the instance. Perform the following steps to recover the instance:

1. Completely shutdown the instance.

2. Perform step 1 of Section 17.3.3 to unpack the latest image backup of the instance.

3. Perform steps 3, 4 and 5 of Section 17.3.3 to register and configure the instance.

   If the instance has any dependencies on other instances of Oracle Application Server, the other instances must be up and running.

18

Recovery Strategies and Procedures

This chapter describes Oracle Application Server recovery strategies and procedures for different types of failures and outages.

It contains the following topics:

- Recovery Strategies
- Recovery Procedures

18.1 Recovery Strategies

This section describes Oracle Application Server recovery strategies for different types of failures and outages. It contains the following topics:

- Recovery Strategies for Data Loss, Host Failure, or Media Failure (Critical)
- Recovery Strategies for Process Failures and System Outages (Non-Critical)

18.1.1 Recovery Strategies for Data Loss, Host Failure, or Media Failure (Critical)

This section describes recovery strategies for outages that involve actual data loss or corruption, host failure, or media failure where the host or disk cannot be restarted and are permanently lost. This type of failure requires some type of data restoration before the Oracle Application Server environment can be restarted and continue with normal processing.

The strategies in this section use point-in-time recovery of the middle tier.

Assumptions

The following assumption applies to the recovery strategies in this section:

- No administrative changes were made since the last backup. If administrative changes were made since the last backup, they will need to be reapplied after recovery is complete.

> **See Also:** Appendix E, "Examples of Administrative Changes" to learn more about administrative changes

Determining Which Strategy to Use

Recovery strategies are listed in Table 18–1.

Use the information in this table if you experience data loss, host failure, or media failure in a middle-tier installation. Find the type of loss and follow the recommended procedure.

Table 18–1   Recovery Strategies for Data Loss, Host Failure, and Media Failure in Middle-Tier Instances

Type of Loss	Recovery Strategies
Loss of host	If the host has been lost, you have two options: ■ You can restore to a new host that has the same hostname and IP address. ■ You can restore to a new host that has a different hostname and IP address. In either case, follow the procedure in Section 18.2.2. Note that if the original host had a middle-tier installation and an Infrastructure, you cannot restore the middle-tier to a host with a different hostname or IP address.
Oracle software/binary deletion or corruption	If any Oracle binaries have been lost or corrupted, you must restore the entire middle tier to the same host. Follow the procedure in Section 18.2.1.
Deletion or corruption of configuration files	If you lose any configuration files in the middle-tier Oracle home, you can restore them. Follow the procedure in Section 18.2.3.

18.1.2 Recovery Strategies for Process Failures and System Outages (Non-Critical)

This section describes recovery strategies for process failures and system outages. These types of outages do not involve any data loss, and therefore do not require any files to be recovered. In some cases, failure may be transparent and no manual intervention is required to recover the failed component. However, in some cases, manual intervention is required to restart a process or component. While these strategies do not strictly fit into the category of backup and recovery, they are included in this book for completeness.

Determining Which Strategy to Use

Recovery strategies for process failures and system outages are listed in Table 18–2.

Use this table if you experience a failure or outage on a middle-tier installation. Find the type of outage and follow the recommended procedure. The table contains UNIX commands. You can use the same commands on Windows by inverting the slashes.

Table 18–2   Recovery Strategies for Process Failures and System Outages in Middle-Tier Instances

Type of Outage	How to Check Status and Restart
Host failure—no data loss	**To restart:** 1. Restart the host. 2. Start the middle tier. Refer to Section 3.2.1.

Application Server Control failure	**To check status:**
	`opmnctl status`
	To restart:
	`opmnctl startproc process-type=OC4J_instance_name`

Oracle HTTP Server process failure	**To check status:**
	`opmnctl status`
	To restart:
	`opmnctl startproc ias-component=HTTP_Server`

Table 18–2   (Cont.) Recovery Strategies for Process Failures and System Outages in Middle-Tier Instances

Type of Outage	How to Check Status and Restart
OC4J instance failure	**To check status:**
	`opmnctl status`
	To restart:
	`opmnctl startproc process-type=OC4J_instance_name`
OPMN daemon failure	**To check status:**
	`opmnctl status`
	To restart:
	`opmnctl start`

18.2 Recovery Procedures

This section contains the procedures for performing different types of recovery.

It contains the following topics:

- Restoring a Middle-Tier Installation to the Same Host
- Restoring a Middle-Tier Installation to a New Host
- Restoring Middle-Tier Configuration Files
- Restoring an Oracle Application Server Instance
- Recovering a Portlet Producer Preference Store

18.2.1 Restoring a Middle-Tier Installation to the Same Host

To restore a middle-tier installation to the same host, refer to Section 17.3.4.

18.2.2 Restoring a Middle-Tier Installation to a New Host

This section describes how to restore and recover a middle-tier installation to a new host. You can use this procedure to:

- Restore a middle-tier installation to the same host after the operating system has been reinstalled.
- Restore a middle-tier installation to a new host. The new host may have the same hostname and IP address as the original host, or a different hostname, IP address, or both.

Perform the steps in Section 17.3.3 to restore the image backup, system files, and instance reconfiguration. Note that the middle-tier configuration remains in the same state as the original instance. If the hostname remains the same, run an instance restore to bring the instance to the desired point in time. If the hostname is different, the state cannot be changed since backups of the original host are not valid for a different hostname.

18.2.3 Restoring Middle-Tier Configuration Files

This section describes how to restore the configuration files in a middle-tier Oracle home. Use this procedure when configuration files have been lost or corrupted.

It contains the following tasks:

- Task 1: Stop the Middle-Tier Instance

- Task 2: Restore Middle-Tier Configuration Files

- Task 3: Apply Recent Administrative Changes

- Task 4: Start the Middle-Tier Instance

Task 1: Stop the Middle-Tier Instance

Refer to Section 3.2.2 for instructions.

Task 2: Restore Middle-Tier Configuration Files

Restore all configuration files from your most recent backup. You can perform this task using your own procedure or OracleAS Recovery Manager. For example, to do this using OracleAS Recovery Manager:

- For UNIX systems:

  ```
  bkp_restore.sh -m restore_config -t timestamp
  ```

- For Windows systems:

  ```
  bkp_restore.bat -m restore_config -t timestamp
  ```

  > **See Also:** Chapter 16, "Oracle Application Server Recovery Manager" for more information

Task 3: Apply Recent Administrative Changes

If you made any administrative changes since the last time you did an online backup, reapply them now.

  > **See Also:** Appendix E, "Examples of Administrative Changes" to learn more about administrative changes

Task 4: Start the Middle-Tier Instance

Refer to Section 3.2.1 for instructions.

18.2.4 Restoring an Oracle Application Server Instance

Use the following command to restore an Oracle Application Server instance to a particular point in time:

```
bkp_restore.sh -m restore_instance -t 2006-09-21_06-12-45
```

```
bkp_restore.bat -m restore_instance -t 2006-09-21_06-12-45
```

Before performing a restore operation (`restore_instance` or `restore_config`) on an instance in a cluster, all OC4J processes across the cluster must be stopped. Use the following command to stop the processes:

```
ORACLE_HOME/opmn/bin/opmnctl @cluster stopproc ias-component=OC4J
```

Some OC4J components do not have `ias-component=OC4J`. For these components use the uniqueid value to stop the OC4J process. To determine which components have an uniqueid, use the following command:

```
ORACLE_HOME\opmn\bin\opmnctl @cluster status -fmt %typ%uid%prt -noheaders
```

The following is an example of the output from the command:

```
CUSTOM     | N/A        | ASG
LOGLDR     | N/A        | logloaderd
```

```
OHS         | 1500577870 | HTTP_Server
performance | 1500577873 | performance_server
messaging   | 1500577874 | messaging_server
```

Stop all the OC4J processes, for which the second column (uid) value is not N/A, with the following command:

```
ORACLE_HOME\opmn\bin\opmnctl @cluster stopproc uniqueid=1500577865

opmnctl: stopping opmn managed processes...
```

After the restore operation completes, use the following command to restart the OC4J processes across the cluster:

```
ORACLE_HOME/opmn/bin/opmnctl @cluster startproc ias-component=OC4J
```

For components that use uniqueid, you can restart their process by using the appropriate ias-component value or by using the following command:

```
opmnctl startall
```

18.2.5 Recovering a Portlet Producer Preference Store

To recover the JPS portlet producer preference store, perform the following steps:

1. Stop the OC4J instance on which the container runs.

2. Run the migration tool to restore data from the source backup file to the destination preference store. For example:

```
java -classpath
  wsrp-container.jar:cache.jar:saaj-api.jar:orasaaj.jar:ojdbc14.jar
  oracle.portlet.server.containerimpl.PersistenceMigrationTool
-sourceType file
-destType db
-sourcePath /tmp/portletbkp
-destUsername p1
-destPassword p1
-destDatabase portaldb.mycompany.com:1521:orcl
```

3. Start the OC4J instance on which the portlet container runs.

For more information about the JPS portlet migration utility, refer to *Oracle WebCenter Framework Developer's Guide*.

To recover the PDK-Java portlet producer preference store, perform the following steps:

1. Stop the OC4J instance on which the container runs.

2. Run the migration tool to restore data from the source backup file to the destination preference store. For example:

```
java -classpath $ORACLE_HOME/portal/jlib/pdkjava.jar
  oracle.portal.provider.v2.preference.MigrationTool
 -mode filetodb
 -remap locale
 -countries AR,MX
 -pref1UseHashing true
 -pref1RootDirectory /tmp/portletbkp
 -pref2User portlet_prefs
 -pref2Password portlet_prefs
 -pref2URL jdbc:oracle:thin:@myserver.mydomain.com:1521:mysid
```

3. Start the OC4J instance on which the portlet container runs.

For more information on the PDK-Java migration and upgrade utility, refer to *Oracle WebCenter Framework Developer's Guide*.

19

Troubleshooting OracleAS Recovery Manager

This chapter describes common problems that you might encounter when using OracleAS Recovery Manager, and explains how to solve them. It contains the following topic:

- Problems and Solutions

19.1 Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

- Receiving Missing Files Messages During restore_config Operation
- Failure Due to Loss or Corruption of OPMN.XML File
- Timeout Occurs While Trying to Stop Processes Using the "opmnctl stopall" Command

19.1.1 Receiving Missing Files Messages During restore_config Operation

A `restore_config` operation generates missing file messages.

Problem

During a restore_config operation, you receive messages indicating that files are missing, for example:

```
Could not copy file C:\Product\OracleAS\Devkit_1129/testdir/ to
C:\Product\OracleAS\Devkit_1129\backup_restore\cfg_bkp/2006-12-01_03-26-22.
```

Solution

During a `restore_config` operation, a temporary configuration backup is taken so that, if the restore fails, the temporary backup can be restored, returning the instance to the same state as before the restore.

If some files are deleted (including file or directories specified in config_misc_files.inp) before a restore operation, then, during the temporary backup, messages are displayed indicating that certain files are missing. These error or warning messages should be ignored since the missing files are restored as part of the `restore_config` operation.

19.1.2 Failure Due to Loss or Corruption of OPMN.XML File

The loss or corruption of the opmn.xml file is causing a failure.

Problem

The loss or corruption of the opmn.xml file caused the following error:

```
ADMN-906025
Base Exception:
The exception, 100999, occurred at Oracle Application Server instance
"J2EE_1123.stad.oracle.com"
```

Resolution

Perform the following steps to restore the opmn.xml file:

1. Run the following command:

   ```
   bkp_restore.bat -m restore_config -t timestamp
   ```

2. If that command fails, stop the OC4J processes.

3. Rerun the following command:

   ```
   bkp_restore.bat -m restore_config -t timestamp
   ```

19.1.3 Timeout Occurs While Trying to Stop Processes Using the "opmnctl stopall" Command

During `backup_instance_cold`, `backup_instance_cold_incr`, and `restore_instance` operations, a timeout may occur while trying to stop processes using the `opmnctl stopall` command.

Problem:

During some operations involving the backup or restore of a server instance, a timeout may occur while trying to stop processes using the `opmnctl stopall` command. This can occur because of heavy machine load or a process taking a long time to shut down. Under these conditions, you may receive an error message similar to the following:

```
Oracle Application Server instance backup failed.
Stopping all opmn managed processes ...

Failure : backup_instance_cold_incr failed

Unable to stop opmn managed processes !!!
```

Solution:

Running `opmnctl stopall` a second time should resolve this problem.

Part VI

Appendixes and Glossary

This part contains the following appendixes and a glossary:

- Appendix A, "Managing and Configuring Application Server Control"
- Appendix B, "Oracle Application Server Command-Line Tools"
- Appendix C, "URLs for Components"
- Appendix D, "Oracle Application Server Port Numbers"
- Appendix E, "Examples of Administrative Changes"
- Appendix F, "Supplementary Procedures for Configuring LDAP-Based Replicas"
- Appendix G, "Viewing Oracle Application Server Release Numbers"
- Appendix H, "Troubleshooting Oracle Application Server"
- "Glossary"

A

Managing and Configuring Application Server Control

When you install Oracle Application Server, the installation procedure automatically starts Oracle Enterprise Manager 10*g* Application Server Control and its related processes. You can then immediately start using Application Server Control to manage the application server components.

You can also control and configure Application Server Control. For example, you can start and stop Application Server Control, change Application Server Control password, and configure security for Application Server Control.

This appendix covers how to manage and configure Application Server Control. It contains the following topics:

- Starting and Stopping Application Server Control
- Changing the Application Server Control Administrator Password
- Configuring Security for Application Server Control
- Configuring Logging for Application Server Control
- Enabling Enterprise Manager Accessibility Mode
- Managing the Active Application Server Control

A.1 Starting and Stopping Application Server Control

For 10*g* Release 3 (10.1.3.4), Application Server Control is deployed as a standard J2EE application. The Application Server Control application (`ascontrol`) is deployed automatically on every OC4J instance you create.

As a result, you can start and stop the `ascontrol` application from the command line, using the procedure described in Section 3.3.1.

You can also stop and restart the active `ascontrol` application from Application Server Control; however, unlike other J2EE applications that you deploy on this release, there are some restrictions when starting and stopping the `ascontrol` application from Application Server Control:

- If you are managing one standalone OC4J instance, then you cannot stop, start, or restart the `ascontrol` application from Application Server Control. If you stopped the `ascontrol` application, you would be unable to display or use Application Server Control.
- If you are in clustered environment, where you are managing multiple OC4J instances, then you can use the Cluster Topology page to start, stop, or restart the

active `ascontrol` application. However, Enterprise Manager displays a warning that describes the implications of stopping the active `ascontrol` application.

The active `ascontrol` application represents Application Server Control currently being used to manage your Oracle Application Server environment. If you stop the active `ascontrol` application, you will no longer be able to use Application Server Control until the application is started.

You cannot, however, use Application Server Control to start a second `ascontrol` application. If you attempt to do so, Application Server Control displays an error message stating that an active ascontrol application is already running.

Note that the OC4J instance used to deploy the active `ascontrol` application is called the **Administration OC4J instance**. OC4J instances other than the Administration OC4J instance are called remote OC4J instances. In most cases, there is no need to start the `ascontrol` in a remote OC4J instance.

However, there is a scenario where the `ascontrol` application in a remote OC4J instance must be running. For more information, see "Starting ascontrol When Viewing Remote Log Files" in the Application Server Control online help.

A.1.1 Verifying That Application Server Control Is Running

You can verify that Application Server Control is started by pointing your browser to the Application Server Control URL:

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

For example:

```
http://mgmthost.acme.com:7777/em
```

To locate the Application Server Control port number, use the following command and check the number for HTTP_Server:

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

> **See Also:** Section 2.3.1, "Displaying the Application Server Control"

A.2 Changing the Application Server Control Administrator Password

To use Application Server Control, you must have an Application Server Control administrator account. The privileges you have when managing your environment are based on the user account and password you use to log in to Application Server Control.

When you install Oracle Application Server, a default super administrator account is created. This account is called `oc4jadmin` and you set the password for this account during the Oracle Application Server installation. You can use the oc4jadmin account to log in to Application Server Control for the first time. Then, you can create additional administration user accounts to use for everyday administration tasks.

> **See Also:** Section 2.3.2, "Creating Administration Accounts for You and Your Team"

Regardless of the user account you use to log in to Application Server Control, you can always change the password for your own administrator account. However, there are special considerations when changing the `oc4jadmin` password.

For more information, see the following sections:

- Changing Your Own Administrator Account Password
- About the oc4jadmin Account
- Changing the oc4jadmin Password for the Administration OC4J Instance
- Using Application Server Control to Change the oc4jadmin Password for a Remote OC4J Instance
- Using the Command Line to Change the oc4jadmin Password for a Remote OC4J Instance

A.2.1 Changing Your Own Administrator Account Password

To change your own administrator account:

1. Log in to Application Server Control using your administrator username and password.

2. Navigate to the Application Server home page and select **Setup** at the top of the page.

    Application Server Control displays the Password page. Note that the **User** field on this page identifies which account you are modifying. If you are modifying the oc4jadmin user account, refer to Section A.2.3 for more information.

3. Enter your current administrator password, the new password, and the new password again for confirmation.

    To provide additional security, the new password:

    - Must contain at least five characters, but not more than 30 characters.
    - Must begin with an alphabetic character. It cannot begin with a number, the underscore (_), the dollar sign ($), or the number sign (#).
    - At least one of the characters must be a number.
    - Can contain only numbers, letters, and the following special characters: US dollar sign ($), number sign (#), or underscore (_).
    - Cannot contain any Oracle reserved words, such as VARCHAR.

    Note that these restrictions are enforced by Application Server Control and Oracle Universal Installer; they are not enforced by the OC4J system-jazn.xml or application-based security configuration files.

4. Click **OK** to reset the password.

    The next time you log in, you must use the new password.

A.2.2 About the oc4jadmin Account

The default oc4jadmin administration user account serves two distinct purposes in the Oracle Application Server environment. Each is described in the following sections:

- Using the oc4jadmin Account to Log In for the First Time
- Using the oc4jadmin Account for Administration Credentials

A.2.2.1 Using the oc4jadmin Account to Log In for the First Time

During the Oracle Application Server installation, you must define a password for the oc4jadmin account. You can then use the oc4jadmin account to log in to Application Server Control for the first time.

The oc4jadmin user account is automatically assigned a set of administrative roles that allow users who log in with this account to manage and configure all aspects of the OC4J instance. However, the oc4jadmin account should not be used for everyday administration tasks. Instead, you should create additional administration accounts for that purpose.

> **See Also:** Section 2.3.2, "Creating Administration Accounts for You and Your Team"

A.2.2.2 Using the oc4jadmin Account for Administration Credentials

In addition to serving as the initial login account for Application Server Control, the oc4jadmin account is also used by Application Server Control software when it performs administration tasks. Specifically, when you perform configuration tasks on an OC4J instance, Application Server Control connects to the instance using the oc4jadmin user account and password. In this context, the oc4jadmin user and password are referred to as the **administration credentials**.

Specifically, when you click the name of an OC4J instance on the cluster topology page, Application Server Control uses the administration credentials to connect to the OC4J instance. Application Server Control uses these credentials to connect to both the local administration OC4J, as well as the other, remote OC4J instances in the cluster. This concept is illustrated in Figure A–1.

Figure A–1   How Application Server Control Uses the Administration Credentials to Manage OC4J Instances in a Cluster
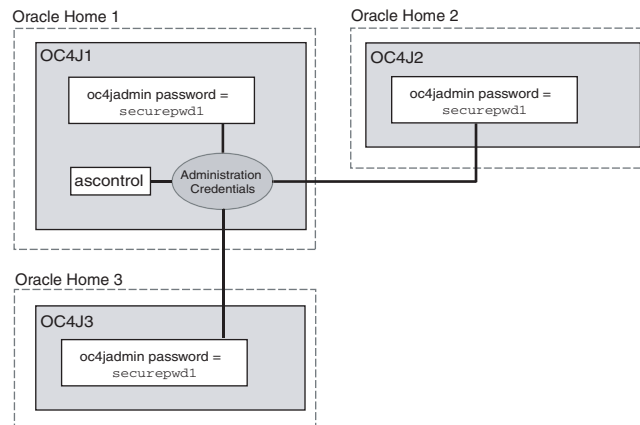
There is only one set of administration credentials for each cluster. By default, the oc4jadmin account and password that you defined for the administration OC4J (Oracle Home 1 in Figure A–1) are stored as the administration credentials for the cluster.

As a result, each OC4J instance in your cluster must have an oc4jadmin user account and the password for the oc4jadmin account must be the same as the password defined for the administration OC4J. Otherwise, the administration credentials defined for the cluster will not work and Application Server Control will not be able to connect to the OC4J instance.

You can later change the password for the `oc4jadmin` account, but if you are managing a cluster of Oracle Application Server instances, you must use caution when changing the `oc4jadmin` password.

A.2.3 Changing the oc4jadmin Password for the Administration OC4J Instance

The procedure for changing the `oc4jadmin` password for the administration OC4J is the same as the procedure for changing your own administrator password. Simply log in using the `oc4jadmin` user name and password, and then click **Setup**.

However, changing the `oc4jadmin` password can have implications on certain operations you perform from Application Server Control. This is because the `oc4jadmin` username and password are used as the administration credentials for the cluster topology. For more information, see Section A.2.2.2.

When you change the `oc4jadmin` password by clicking **Setup** on any page in Application Server Control, you are changing the password for the `oc4jadmin` account in the administration OC4J instance only.

Changing the password through the Setup link does not change the `oc4jadmin` password used by any remote OC4J instances. A remote OC4J instance is any OC4J instance in a cluster topology that is not hosting the active Application Server Control.

As a result, if you change the `oc4jadmin` password by using the Setup link in Application Server Control so it is different from the `oc4jadmin` password defined in the remote OC4J instances, you will no longer be able to navigate to those instances from the Cluster Topology page. Enterprise Manager displays the following error message when you attempt to display the home page of an OC4J instance where the administration credentials are not valid:

```
Unable to make a connection to OC4J instance instance_name on Application
Server application_server_name. A common cause for this failure is an
authentication error. The administrator password for each OC4J instance in the
Cluster must be the same as the administrator password for the OC4J instance on
which Application Server Control is running.
```

To remedy this problem, you must change the administration credentials for the remote OC4J instance so it matches the administration credentials for the administration OC4J.

A.2.4 Using Application Server Control to Change the oc4jadmin Password for a Remote OC4J Instance

If you are managing multiple OC4J instances in a cluster topology, you can use the Setup link at the top of the Cluster Topology page to change the password for the Administration OC4J, which hosts the `ascontrol` application.

However, to change the `oc4jadmin` password of a remote OC4J instance in a cluster topology, you must use the following procedure.

Note that this procedure assumes that:

- The `oc4jadmin` account is being used as the administration credentials for the cluster.

- The `oc4jadmin` password is currently the same for both the administration OC4J and the remote OC4J instance. If it is not, you will not be able to use Application Server Control to change the remote `oc4jadmin` password; instead, use the procedure described in Section A.2.5.

To use Application Server Control to change the `oc4jadmin` password for a remote OC4J instance:

1. From the Cluster Topology page, click the name of the remote OC4J instance you want to modify.

   Be sure that you are selecting a remote OC4J instance and not the Administration OC4J that hosts the active `ascontrol` application.

   > **See Also:** Section 2.3.3.1, "Viewing the Cluster Topology and Locating the Active Application Server Control"

   Enterprise Manager displays the OC4J home page for the selected remote instance.

2. Click **Administration** to display the list of administration tasks you can perform on the selected OC4J instance.

3. Click the task icon in the **Security Providers** row of the table.

4. On the Security Providers page, click **Instance Level Security**.

5. On the Instance Level Security page, click **Realms**.

6. In the jazn.com row of the Results table, click the number (for example, 3) in the Users column.

   Enterprise Manager displays the list of users defined for the selected security provider.

7. Click **oc4jadmin** to modify the `oc4jadmin` user account.

8. Use the password fields on the User page to change the password of the `oc4jadmin` account for this remote OC4J instance, and then click **Apply**.

9. Return to the Cluster Topology page and restart the remote OC4J instance.

A.2.5 Using the Command Line to Change the oc4jadmin Password for a Remote OC4J Instance

To change the oc4jadmin password for a remote account without using Application Server Control:

1. Log in to the computer where the remote OC4J is installed and running.

2. Locate the following configuration file in the Oracle home of the OC4J instance:

   ORACLE_HOME/j2ee/*oc4j_instance_name*/config/system-jazn-data.xml

3. Use a text editor to open the `system-jazn-data.xml` file and locate the following entry for the `oc4jadmin` user:

   ```
   <user>
        <name>oc4jadmin</name>
        <display-name>OC4J Administrator</display-name>
        <guid>41A2E560C96711DABFD08D3BF8B780C4</guid>
        <description>OC4J Administrator</description>
        <credentials>{903}4nlfYYDwaqMJipVbGXuS2ce8egfwBPqp</credentials>
   </user>
   ```

4. Replace the value of the <credentials> element with the new password. Be sure to include an exclamation point character (!) before the password.

   For example:

```
<credentials>!abcdefg1234</credentials>
```

In this example, replace *abcdefg1234* with the actual password you want to use.

The exclamation point ensures that the password will be encrypted in the configuration file.

> **See Also:** "Password Obfuscation in OC4J Configuration Files" in the *Oracle Containers for J2EE Configuration and Administration Guide*

5. Save your changes and exit the `system-jazn-data.xml` file.

6. Restart the OC4J instance.

   For example, use the following Oracle Process Manager and Notification Server (OPMN) commands to stop and then start the Oracle Application Server instance:

   ```
   ORACLE_HOME/opmn/bin/opmnctl stopall
   ORACLE_HOME/opmn/bin/opmnctl startall
   ```

A.3 Configuring Security for Application Server Control

Securing Application Server Control involves securing two types of communication links:

- Browser client to server communication
- Communication between Oracle Application Server components

Enabling security involves some trade-offs: higher security may mean the use of SSL and the need for more processing power and memory. Because of this, security measures should be applied where they are needed, depending on your environment.

The following sections describe how to configure security for the Application Server Control application:

- Securing Communication Between Browser Clients and Web Servers That Host Application Server Control
- Securing Communication Between Components of Oracle Application Server

> **Note:** This section provides an overview of the steps you must perform to secure Application Server Control. For more complete instructions on the security settings and options described in this section, refer to:
>
> - *Oracle Containers for J2EE Security Guide*
> - *Oracle HTTP Server Administrator's Guide*

A.3.1 Securing Communication Between Browser Clients and Web Servers That Host Application Server Control

By default, Application Server Control user credentials are sent (over a corporate network or the internet) in clear text from the browser to the Web server. As such, it is vulnerable to a security attack.

To secure communication between browser clients and Web servers that host Application Server Control, you must encrypt all Application Server Control communication (including Application Server Control user credentials).

In a secure configuration, browser clients connect directly to the Administration OC4J instance over HTTPS to access Application Server Control. This is the recommended configuration in both OC4J standalone installations and in Oracle Application Server environments.

The following procedure describes how to configure the Administration OC4J instance to serve Application Server Control clients using HTTPS:

Task 1: Create a Keystore and SSL Certificate for the Administration OC4j

To create a keystore and SSL certificate for the Administration OC4J instance, take the following steps:

1. Stop the Administration OC4J instance.

2. Create a keystore with an RSA private/public keypair using the `keytool` executable. This creates an SSL certificate that OC4J can use for secure HTTP communication with browser clients. The `keytool` executable is located in the `ORACLE_HOME`/jdk/bin directory. Use the following command:

   ```
   keytool -genkey -keyalg "RSA" -keystore keystore -storepass passwd -validity
   days
   ```

   When you are prompted for a `key` password, press Return, rather than entering a different password. The `key` password is used to protect the private key of the generated key pair. You must use the same password as the keystore password for SSL to work properly.

   > **See Also:**
   >
   > - "Using Keys and Certificates with OC4J and Oracle HTTP Server" in the *Oracle Containers for J2EE Security Guide*
   >
   > - The description of the JSSE `keytool` command, which is available on the Sun Microsystems Web site:
   >
   >   `http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html`

Task 2: Unbind the ascontrol Application from the Non-Secure Web Site

To unbind the ascontrol Web application from the default non-secure Web site, take the following steps:

1. Edit the configuration file for the Web site where the Application Server Control (`ascontrol`) Web module is bound. By default, the file is:

   ```
   (UNIX) ORACLE_HOME/j2ee/Admin_OC4J_instance_name/config/default-web-site.xml
   (Windows) ORACLE_HOME\j2ee\Admin_OC4J_instance_name\config\default-web-site.xml
   ```

2. Remove the `<web-app>` element that binds the `ascontrol` application. For example, remove the following line:

   ```
   <web-app application="ascontrol" name="ascontrol" root="/em"
   load-on-startup="true" ohs-routing="true" />
   ```

3. Save and close the file.

Task 3: Create a New HTTPS Web Site for the ascontrol Application

Create a new Web site for the Application Server Control (`ascontrol`) application by creating a new configuration file in the Administration OC4J instance that uses HTTPS. Take the following steps:

1. Copy an existing `*-web-site.xml` file in the *ORACLE_HOME*/j2ee/*Admin_OC4J_instance_name*/config directory to create a new Web site. For example, copy `default-web-site.xml` to `ascontrol-web-site.xml`.

2. Make the following changes to the `<web-site>` element of the newly created `ascontrol-web-site.xml` file:

   ■ Set the display name of the Web site to `ASControl Secure HTTP Web Site` by modifying the `display-name` attribute.

   ■ Configure the Web site to use HTTPS by setting the `protocol` attribute to `http`, and by setting the `secure` attribute to `true`.

   ■ Configure the port that browser clients will use to access the Application Server Control Web site, by setting an new port number in the `port` attribute. For example, set `port` to 1156.

   ■ Add an `<ssl-config>` element with its required `keystore` and `keystore-password` properties to reference the keystore you created in the previous task.

   ■ Modify the `path` attribute of the `<access-log>` element to point to a new log file to store the new Web site's access log.

3. Bind the `ascontrol` Web module to this Web site by:

   ■ Setting the `application` and `name` attributes of the `<default-web-app>` element within the `<web-site>` element to `ascontrol`

   ■ Setting the `root` attribute of the `<default-web-app>` element to `"/ "`

   ■ Removing all other `<web-app>` elements within the `<web-site>` element

The following excerpt of a Web site configuration file, named `ascontrol-web-site.xml`, is an example of a dedicated Web site for the `ascontrol` Web application:

```
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/web-site-10
_0.xsd"
          port="1156" protocol="http" secure="true"
          display-name="ASControl Secure HTTP Web Site""
                       schema-major-version="10" schema-minor-version="0" >

    <default-web-app application="ascontrol" name="ascontrol" root="/ " />
    <access-log path="../log/ascontrol-web-access.log" split="day" />
    <ssl-config keystore="private/OracleAS_2/jdk/bin/mykeystore"
             keystore-password="welcome"/>
</web-site>
```

Note that the value of the `keystore` attribute is either an absolute path or a path relative to the location of the `*-web-site.xml` file.

In this example, Application Server Control users access the console by accessing the following URL:

```
https://hostname:1156
```

Task 4: Register the New ascontrol HTTPS Web Site

Register the new Web site in the Administration OC4J instance:

1. Locate the `server.xml` file in the *ORACLE_HOME*/j2ee/*Admin_OC4J_instance_name*/config directory.

2. Add a `<web-site>` element to the `<application-server>` element pointing to the new `ascontrol-web-site.xml` file (the `path` attribute is absolute or relative to the location of the `server.xml` file). For example:

```
<web-site path="./ascontrol-web-site.xml" />
```

3. If the Administration OC4J instance is in a clustered environment, register the new Web site with OPMN by modifying the following file:

```
(UNIX) ORACLE_HOME/opmn/conf/opmn.xml
(Windows) ORACLE_HOME\opmn\conf\opmn.xml
```

Locate the `<ias-component>` element for the Administration OC4J (under ias-component ID `OC4J` and the process-type ID that is equal to the name of the Administration OC4J). Add a new `<port>` element for the new Web site in the Administration OC4J section. For example:

```
<ias-instance id="yellow.stadm21.ora.com" name="yellow.stadm21.ora.com">
 . . .
        <ias-component id="OC4J">
            <process-type id="home" module-id="OC4J" status="enabled">
 . . .
                <port id="default-web-site" range="8989" protocol="http"/>
                <port id="secure-web-site" range="1156" protocol="https"/>
                <port id="rmi" range="12401-12500"/>
                <port id="jms" range="12601-12700"/>
                <process-set id="default_group" numprocs="1"/>
 . . .
            </process-type>
        </ias-component>
```

In this example, the Oracle Application Server instance name is `yellow.stadm21.ora.com` and the Administration OC4J instance name is `home`.

Task 5: Start the Administration OC4J Instance

In an Oracle Application Server environment, reconfigure OPMN with the new opmn.xml file by reloading the opmn.xml file and starting the Administration OC4J instance. Use the following commands:

- UNIX:

```
ORACLE_HOME/opmn/bin/opmnctl reload
ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=OC4J
```

- Windows:

```
ORACLE_HOME\opmn\bin\opmnctl reload
ORACLE_HOME\opmn\bin\opmnctl startproc ias-component=OC4J
```

> **See Also:**
>
> - "Using SSL with OC4J in Oracle Application Server" in the *Oracle Containers for J2EE Security Guide*
>
> - "Using SSL with Standalone OC4J" in the *Oracle Containers for J2EE Security Guide*

A.3.2 Securing Communication Between Components of Oracle Application Server

Depending on your environment, you may choose to secure communication between components of Oracle Application Server. Each communication link is independent of the other, so you have complete flexibility over which links you want to secure and which you do not. You have the option to:

- Encrypt communication between the Administration OC4J and remote OC4J instances (thereby encrypting the `oc4jadmin` password of the remote OC4J).

- Secure OPMN so that only trusted Oracle Application Server instances can join the cluster.

A **remote OC4J instance** is an OC4J instance that is managed remotely by Application Server Control. The remote OC4J instance may reside in the same Oracle home as the Administration OC4J, in a different Oracle home and on the same host, or on a different host.

The following sections describe these topics:

- Securing Communication Between the Administration OC4J and Remote OC4J Instances

- Securing OPMN Communication in an Oracle Application Server Cluster

A.3.2.1 Securing Communication Between the Administration OC4J and Remote OC4J Instances

In an Oracle Application Server environment, when Application Server Control is used to manage OC4J instances other than the Administration OC4J, it uses the Remote Method Invocation (RMI) protocol to establish a JMX connection with remote OC4J instances. In establishing a JMX connection to a remote OC4J, Application Server Control authenticates itself by sending the `oc4jadmin` user credentials of the remote OC4J. By default, this communication happens in clear text.

Use the Secure Remote Method Invocation (ORMIS) protocol to secure communication between the Administration OC4J and remote OC4J instances.

The following procedure describes the tasks you must perform to enable RMIS for the Administration OC4J instance, as well as each of the OC4J instances you are managing with Application Server Control.

Note that this procedure is necessary only in a managed Oracle Application Server environment that you have installed with Oracle Universal Installer and the Oracle Application Server installation procedure.

> **See Also:** For complete information about securing ORMI connections for deployment and management, as well as for instructions on configuring ORMIS in a standalone environment, see the *Oracle Containers for J2EE Security Guide*.

Task 1: Configure Each OC4J Instance with an RMIS Port

Configure a Secure RMI port on the Administration OC4J instance and on each remote OC4J instance that is being managed by Application Server Control:

1. Create a keystore with an RSA private/public keypair using the `keytool` executable. This creates an SSL certificate that OC4J can use for secure HTTP communication with browser clients. The `keytool` executable is located in the *ORACLE_HOME*/jdk/bin directory. Use the following command:

   ```
   keytool -genkey -keyalg "RSA" -keystore keystore -storepass passwd -validity days
   ```

When you are prompted for a `key` password, press Return, rather than entering a different password. The `key` password is used to protect the private key of the generated key pair. You must use the same password as the keystore password for SSL to work properly.

> **See Also:**
> - "Using Keys and Certificates with OC4J and Oracle HTTP Server" in the *Oracle Containers for J2EE Security Guide*
>
> - The description of the JSSE `keytool` command, which is available on the Sun Microsystems Web site:
>
>   `http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html`

2. Locate the `rmi.xml` configuration file for the OC4J instance.

   The file is typically located in the following location; however, you can verify its location by checking the value of the `<rmi-config>` element in the `server.xml` file for the OC4J instance:

   ```
   (UNIX) ORACLE_HOME/j2ee/instance_name/config/rmi.xml
   (Windows) ORACLE_HOME\j2ee\instance_name\config\rmi.xml
   ```

3. Open the `rmi.xml` file with a text editor and add the `<ssl-config>` element to the contents of the file.

4. Use the `<ssl-config>` element to specify the path to the keystore you created in step 1 and the keystore password. For example:

   ```
   <ssl-config keystore="path_to_keystore" keystore-password="keystore_pwd" />
   ```

5. Use the `ssl-port` attribute in the `<rmi-server>` element to specify the SSL listener port. For example:

   ```
   <rmi-server ... port="23791" ssl-port="23943" ... >
   ```

Task 2: Distribute the SSL Certificate of Each Remote OC4J Instance to the Administrative OC4J Instance

You must distribute the SSL certificate of each remote OC4J instance to the Administration OC4J instance. You can do this either by having each remote Administration OC4J instance use an SSL certificate that has been signed by a certificate authority that is trusted by the Administration OC4J's keystore or by importing the SSL certificate of each remote OC4J instance into the Administration OC4J's keystore.

To import the SSL certificate of each remote OC4J instance into the Administration OC4J's keystore, take the following steps for each remote OC4J instance:

1. From the remote OC4J Oracle home, use the `keytool` command to export the OC4J SSL certificate, which contains the RSA public key. This step places the certificate into a file that is accessible to the Administration OC4J.

   ```
   keytool -export -file cert_file_name -keystore keystore_file_name
   ```

2. Import the OC4J SSL certificate into the Administration OC4J keystore, by executing the following command from the Administration OC4J Oracle home:

   ```
   keytool -import -file cert_file_name -keystore keystore_file_name
   ```

Task 3: Configure OPMN to Enable RMIS

Perform the following steps for each Oracle Application Server instance that hosts an OC4J instance in your environment:

1. Locate the following configuration file in the Oracle home:

   ```
   (UNIX) ORACLE_HOME/opmn/conf/opmn.xml
   (Windows) ORACLE_HOME\opmn\conf\opmn.xml
   ```

2. Open the `opmn.xml` file with a text editor and add a new `<port>` element for the RMIS protocol for each OC4J instance defined in the `opmn.xml` file:

   ```
   <port id="rmis" range="12701-12800"/>
   ```

Task 4: Configure the Administration OC4J Instance for Secure RMIS Connection Policy

1. Locate the OPMN configuration file in the Oracle home where the Administration OC4J is installed:

   ```
   (UNIX) ORACLE_HOME/opmn/conf/opmn.xml
   (Windows) ORACLE_HOME\opmn\conf\opmn.xml
   ```

2. Locate the `<ias-component>` element for the Administration OC4J (under ias-component ID `OC4J` and the process-type ID equal to the name of the Administration OC4J). Add the following property to the java-options start parameters for the Administration OC4J in the `opmn.xml` file:

   ```
   oracle.oc4j.jmx.internal.connection.protocol
   ```

   Application Server Control uses this property to determine when to use the secure RMI protocol to communicate with remote OC4J instances.

Table A–1 shows the values you can assign to this property depending on the level of security you want to enforce in your environment.

The following example shows a typical configuration for the `<ias-component>` element of the Administration OC4J with the RMIS property set to RMIS.

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="-server
          -Doracle.oc4j.jmx.internal.connection.protocol=RMIS
          -Djava.security.policy=$ORACLE_HOME/j2ee/home/config/java2.policy
          -Djava.awt.headless=true -Dhttp.webdir.enable=false"/>
      </category>
    </module-data>
  </process-type>
</ias-component>
```

In this example, the name of the Administration OC4J is `home`.

Note that if you want to maintain secure connections when managing all your OC4J instances and applications, you must add the `<ssl-config>` element to the `rmi.xml` file for each OC4J instance you are managing. Otherwise, management connections to the OC4J instance from Application Server Control will either fail or use the non-secure RMI protocol, depending upon the value of the connection protocol property in the `opmn.xml` file for the Administration OC4J instance.

Table A–1   Possible Values for the jmx.internal.connection.protocol Property

Property Value	Description
RMIS_RMI	Use RMIS if available; otherwise use RMI.
	This is the default setting if the RMI connection protocol is not found in the `opmn.xml` file.
RMI_RMIS	Use RMI if available; otherwise use RMIS.
RMIS	Use RMIS; if RMIS is not available, then report a failed connection.
RMI	Use RMI; if RMI is not available, then report a failed connection.

See "Enabling ORMIS for OC4J" in the *Oracle Containers for J2EE Security Guide* for more information.

A.3.2.2 Securing OPMN Communication in an Oracle Application Server Cluster

If your environment includes a cluster topology, you must secure the cluster so that only trusted Oracle Application Server instances can join the cluster. Otherwise, a malicious Oracle Application Server instance can join the cluster and gain process control over the cluster components on the Cluster Topology page.

For more information, see the following sections:

- About Securing OPMN and Why It Is Important
- Securing the Cluster by Distributing a Common Wallet for OPMN SSL Connections

A.3.2.2.1   About Securing OPMN and Why It Is Important During installation, OPMN is configured to use a default wallet containing a default SSL certificate. The default wallet is stored in the following directory in each Oracle Application Server Oracle home:

```
(UNIX) ORACLE_HOME/opmn/conf/ssl.wlt/default
(Windows) ORACLE_HOME\opmn\conf\ssl.wlt\default
```

To secure OPMN effectively, you must replace this default wallet with a new wallet that contains a valid and unique certificate. You must perform this task for each Oracle Application Server instance in a cluster.

In addition, you must use mutual authentication (also referred to as Server and Client Authentication mode) when you are configuring the wallet and certificates for OPMN.

In mutual authentication mode, both the OPMN client and the OPMN server use X.509 certificates to authenticate each other before a connection is made. First, the client authenticates the server by validating its certificate. In return, the server also requires the client to send its certificate to prove its authenticity.

If you do not replace the default wallet in each OPMN in your cluster and ensure that you are using mutual authentication for all instances in your cluster, then any Oracle Application Server instance that uses the default wallet and certificate will be able to join the cluster. After it joins the cluster, the Application Server Control for the instance can start and stop components of the cluster on the Cluster Topology page.

A.3.2.2.2   Securing the Cluster by Distributing a Common Wallet for OPMN SSL Connections
The easiest way to secure OPMN connections between the Oracle Application Server instances in your cluster is to create a new wallet and distribute that wallet to each Oracle Application Server instance in the cluster.

In the following example, a self-signed (root) certificate is used to secure an Oracle Application Server cluster. In a more complex or production environment, you can use certificates obtained from a Certificate Authority.

Use the following example to understand how to set up mutual authentication using a self-signed certificate:

1. Use the `orapki` utility to create a new wallet and a new self-signed (root) certificate for the first Oracle Application Server instance in the cluster topology.

   > **See Also:** Section 11.2, "Performing Certificate Validation and CRL Management with the orapki Utility"

2. Modify the OPMN configuration file (`opmn.xml`) for the instance so that the `<ssl>` element points to the newly created wallet. For example:

```
<ssl enabled="true"
     wallet-file="$ORACLE_HOME/opmn/conf/ssl.wlt/new_wallet"
     wallet-password="welcome2"/>
```

3. Copy the new wallet, which contains the newly created root certificate, to each of the Oracle Application Server instances in the cluster and update the `opmn.xml` file for each instance as described in Step 2.

4. Restart each Oracle Application Server instance in the cluster using the following commands:

   UNIX:

```
ORACLE_HOME/opmn/bin/opmnctl stopall
ORACLE_HOME/opmn/bin/opmnctl startall
```

   Windows:

```
ORACLE_HOME\opmn\bin\opmnctl stopall
ORACLE_HOME\opmn\bin\opmnctl startall
```

5. Use Application Server Control to verify that the cluster is configured correctly:

   a. Log in to Application Server Control and review the contents of the Cluster Topology page.

      Each of the Oracle Application Server instances in the cluster should be listed in the Members section of the page.

   b. Scroll to the Administration section of the page and click **Topology Network Configuration**.

   c. Select any one of the Oracle Application Server instances in the cluster from the **View By** menu near the top of the Topology Network Configuration page.

   d. Scroll to the SSL section of the Topology Network Configuration page and verify that SSL is enabled and the instance is using the new wallet you created earlier in this procedure.

See the *Oracle Process Manager and Notification Server Administrator's Guide* for more information about OPMN and security.

A.4 Configuring Logging for Application Server Control

Application Server Control provides its own set of log files, which you can configure by modifying a configuration file. How you configure logging depends upon whether or not you enable Oracle Diagnostic Logging (ODL).

For more information, see the following sections:

- Enabling and Configuring ODL for the Application Server Control Log File
- Configuring Logging Properties When ODL Is Not Enabled
- Controlling the Number of Entries Retrieved When Searching Log Files

A.4.1 Enabling and Configuring ODL for the Application Server Control Log File

By default, the log file generated for Application Server Control is saved in text format. However, you can configure Application Server Control so its log file will be saved using the Oracle Diagnostic Logging (ODL) format.

When you enable ODL for the Application Server Control log files, the logging and diagnostic information is saved in XML format and each log message is formatted to comply with the ODL standard.

> **See Also:** Chapter 5, "Managing Log Files"

By default, Application Server Control logs information and errors to the following log file in the application server home directory:

```
(UNIX) ORACLE_HOME/j2ee/home/log/ascontrol.log
(Windows) ORACLE_HOME\j2ee\home\log\ascontrol.log
```

After you perform the procedure in Section A.4.1.1, Application Server Control will instead log information and error messages to the following file, which formats the data according to the ODL standard:

```
(UNIX) ORACLE_HOME/sysman/log/log.xml
(Windows) ORACLE_HOME\sysman\log\log.xml
```

Refer to the following sections for more information:

- Section A.4.1.1, "Configuring the Application Server Control Logging Properties to Enable ODL"
- Section A.4.1.2, "About the Application Server Control ODL Logging Properties"
- Section A.4.2, "Configuring Logging Properties When ODL Is Not Enabled"

A.4.1.1 Configuring the Application Server Control Logging Properties to Enable ODL

To configure the Application Server Control to support ODL:

1. Navigate to the following directory in the Oracle Application Server Oracle home:

   ```
   (UNIX) ORACLE_HOME/j2ee/home/applications/ascontrol/ascontrol/WEB-INF/config
   (Windows) ORACLE_HOME\j2ee\home\applications\ascontrol\ascontrol\WEB-INF\config
   ```

2. Use a text editor to edit the following configuration file in the config directory:

   ```
   ascontrollogging.properties
   ```

3. Follow the instructions in the file to replace the default properties with those that are commented by default.

Example A–1 shows the properties in the `emiasconsolelogging.properties` file that enable ODL for the Application Server Control log file.

4. Save and close the `ascontrollogging.properties` file.

5. Restart Application Server Control.

Example A–1   ODL Logging Properties for Application Server Control

```
# To support the ODL log appender, replace the lines above
# with the following and restart EM.  The resulting ODL log files
# will be read by the Log Loader and written to the Log Repository.
#
# log4j.appender.emiaslogAppender=oracle.core.ojdl.log4j.OracleAppender
# log4j.appender.emiaslogAppender.ComponentId=EM
# log4j.appender.emiaslogAppender.LogDirectory=/private/shiphomes/m21_
infra/sysman/log
# log4j.appender.emiaslogAppender.MaxSize=20000000
# log4j.appender.emiaslogAppender.MaxSegmentSize=5000000
```

A.4.1.2 About the Application Server Control ODL Logging Properties

Table A–2 describes the Oracle Diagnostic Logging (ODL) logging properties available in the `emiasconsolelogging.properties` file.

Table A–2   Oracle Diagnostic Logging (ODL) Properties

Property	Description
log4j.appender.emiaslogAppender. LogDirectory	Determines the directory where the `log.xml` file will be saved.
log4j.appender.emiaslogAppender. MaxSize	Determines the maximum amount of disk space to be used by the `log.xml` file and the logging rollover files.
log4j.appender.emiaslogAppender. MaxSegmentSize	Determines the maximum size of the `log.xml` file. When the `log.xml` file reaches this size, a rollover file is created.

When you enable ODL, the resulting `log.xml` file increases in size over time as information is written to the file. The file is designed to reach a maximum size, determined by the `MaxSegmentSize` property described in Table A–2. When the file reaches the predefined maximum size, Application Server Control renames (or rolls) the logging or trace information to a new file name and starts a new log or trace file. This process keeps the log file from growing too large.

To be sure you have access to important log information, Application Server Control will roll over the `log.xml` file until the log file and its rollover files consume a predefined, maximum amount of disk space, determined by the MaxSize property shown in Example A–1. When the log file and its rollover files reach this predefined target, Application Server Control deletes the oldest rollover file.

As a result, you will often see multiple log files in the log directory. The following example shows three Application Server Control rollover files and the current log file in the log directory:

```
log.xml
log1.xml
log2.xml
log3.xml
```

A.4.2 Configuring Logging Properties When ODL Is Not Enabled

If you do not enable ODL, you can still configure the logging properties for Application Server Control by modifying the `ascontrollogging.properties` file:

1. Navigate to the following directory in the Oracle Application Server home directory:

   ```
   (UNIX) ORACLE_HOME/j2ee/home/applications/ascontrol/ascontrol/WEB-INF/config/
   (Windows) ORACLE_
   HOME\j2ee\home\applications\ascontrol\ascontrol\WEB-INF\config\
   ```

2. Use a text editor to edit the following configuration file in the `config` directory:

   ```
   ascontrollogging.properties
   ```

3. Modify the selected logging properties described in Table A–3.

4. Save and close the `ascontrollogging.properties` file.

5. Restart Application Server Control.

Table A–3   Logging Properties When ODL Is Not Enabled

Property	Description
log4j.appender.ascontrollog Appender.File	The location and name of the Application Server Control (ascontrol) application.
log4j.appender.ascontrollog Appender.MaxFileSize	Determines the maximum amount of disk space to be used by the ascontrol application log file and its rollover log files.
log4j.appender.ascontrollog Appender.MaxBackupIndex	Indicates how many times Application Server Control will rollover its log file to a new file name before deleting the oldest rollover log file.

A.4.3 Controlling the Number of Entries Retrieved When Searching Log Files

To help you diagnose application errors or performance problems, Enterprise Manager provides a mechanism for searching the OC4J log files from Application Server Control.

You can search not only all the OC4J instance log files associated with the current Oracle Application Server instance, but the log files associated with the applications you deploy on the instance.

> **See Also:**
>
> - Section 5.1, "Listing and Viewing Log Files with Application Server Control"
> - "Searching OC4J Log Files" in the Application Server Control online help

On the Search Logs page in Application Server Control, you can enter a set of search criteria and Enterprise Manager will return a list of log file entries that match your criteria. By the default, Enterprise Manager will return a maximum of 5,000 log file entries that you can browse or filter for more information.

To retrieve more than 5,000 entries during a log file search, or to limit searches so they return less than 5,000 entries, use the following procedure:

1. Use a text editor to open the `emiaslogviewer.xml` configuration file, which is located in following directory in the Oracle home of the administration OC4J:

```
(UNIX) ORACLE_HOME/j2ee/home/applications/ascontrol/ascontrol/WEB-INF/config
(Windows) ORACLE_HOME\j2ee\home\applications\ascontrol\ascontrol\WEB-INF\config
```

2. Modify the following configuration file in the `emiaslogviewer.xml` file:

    ```
    <LogViewerConfig maxEntries="5000">
    ```

A.5 Enabling Enterprise Manager Accessibility Mode

The following sections provide information on the benefits of running Enterprise Manager in accessibility mode, as well as instructions for enabling accessibility mode:

- Making HTML Pages More Accessible
- Providing Textual Descriptions of Enterprise Manager Charts
- Modifying the uix-config.xml File to Enable Accessibility Mode

A.5.1 Making HTML Pages More Accessible

Enterprise Manager takes advantage of user interface development technologies that improve the responsiveness of some user operations. For example, when you navigate to a new record set in a table, Enterprise Manager does not redisplay the entire HTML page.

However, this performance-improving technology is generally not supported by screen readers. When you enable accessibility mode, you disable this feature, and as a result, make the Enterprise Manager HTML pages more accessible for disabled users.

A.5.2 Providing Textual Descriptions of Enterprise Manager Charts

Throughout Enterprise Manager, 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 Enterprise Manager to provide a complete textual representation of each performance chart. When you enable accessibility mode, Enterprise Manager displays a small icon for each chart that can be used as a drill-down link to the textual representation.

Figure A–2 shows an example of the icon that appears below each chart after you enable accessibility mode.

Figure A–2   Icon Representing the Textual Representation of a Chart
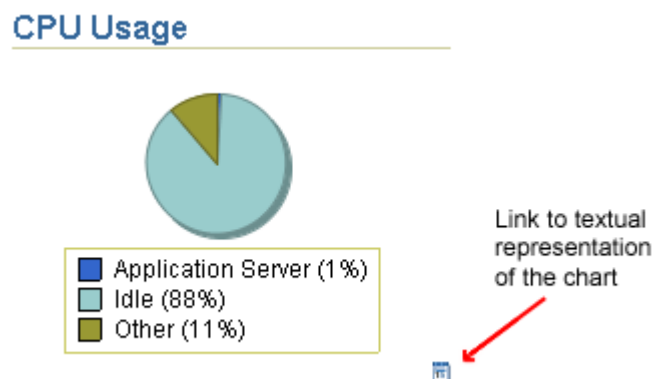
A.5.3 Modifying the uix-config.xml File to Enable Accessibility Mode

To modify the `uix-config.xml` configuration file:

1. Locate the `uix-config.xml` configuration file in the Oracle Application Server home directory:

   ```
   (UNIX) ORACLE_HOME/j2ee/home/applications/ascontrol/WEB-INF
   (Windows) ORACLE_HOME\j2ee\home\applications\ascontrol\WEB-INF
   ```

2. Open the `uix-config.xml` file using a text editor and locate the following entry:

   ```
   <!-- An alternate configuration that disables accessibility features  -->
   <default-configuration>
     <accessibility-mode>inaccessible</accessibility-mode>
   </default-configuration>
   ```

3. Change the value of the `accessibility-mode` property from `inaccessible` to `accessible`.

4. Save and close the file.

5. Restart Application Server Control.

A.6 Managing the Active Application Server Control

Use the following sections to learn more about managing and configuring the active Application Server Control:

- About the Active Application Server Control

- Best Practices for Managing the Active Application Server Control

- Stopping an Instance of ascontrol and Preventing the Application from Starting

- Identifying and Configuring a New Active Application Server Control

- Accessing the Administration OC4J Directly Through HTTP

- Publishing Application Server Control to a Separate Web Site in the Same OC4J Instance

A.6.1 About the Active Application Server Control

By default, each OC4J instance in an Oracle Application Server 10*g* Release 3 (10.1.3.4) cluster contains an `ascontrol` application, which represents an instance of Application Server Control.

However, only one `ascontrol` application should be running in a cluster. The other instances of the application should be stopped or disabled. When you create a new OC4J instance, the new instance includes an `ascontrol` application, which is deployed to the OC4J instance. However, by default, a setting in the `server.xml` configuration file, which defines the characteristics of the OC4J instance, prevents the ascontrol application from starting automatically when the OC4J instance is started.

> **See Also:** Section A.6.3, "Stopping an Instance of ascontrol and Preventing the Application from Starting"

Whenever you log in to Application Server Control, only one `ascontrol` application is being used to manage your cluster. That `ascontrol` application is called the active Application Server Control.

❖ To identify the active Application Server Control, navigate to the Cluster Topology page, click **Expand All** to view all components of the cluster, and then locate the `ascontrol` application that is identified by the active Application Server Control icon.

The OC4J instance that hosts the active Application Server Control is referred to as the **administration OC4J**.

A.6.2 Best Practices for Managing the Active Application Server Control

Table A–4 lists a set of best practice guidelines that you should consider for managing the active Application Server Control.

Table A–4   Best Practices for Managing the Active Application Server Control

Recommendation	Explanation	More information
Select one OC4J instance that will serve as your administration OC4J.	Do not use this instance to host any of your deployed applications. Instead, use it exclusively to host Application Server Control and to create additional OC4J instances to deploy your applications. Note that if you are using Java SSO, you can also use the administration OC4J to host the active `javasso` application.	Section 6.2, "Adding and Deleting OC4J Instances" Section 6.6, "Configuring Instances to Use OC4J Java Single Sign-On"
Shut down or disable instances of the `ascontrol` application that are not deployed to the administration OC4J instance.	Only the active Application Server Control should be running.	Section A.6.3, "Stopping an Instance of ascontrol and Preventing the Application from Starting"
Isolate Application Server Control so it uses its own Web site or URL.	Only your Oracle Application Server administrators need access to Application Server Control so it makes sense to use a separate URL and port number to access the console.	Section A.6.5, "Accessing the Administration OC4J Directly Through HTTP" Section A.6.6, "Publishing Application Server Control to a Separate Web Site in the Same OC4J Instance"

A.6.3 Stopping an Instance of ascontrol and Preventing the Application from Starting

If more than one `ascontrol` application is running in your cluster topology, the following warning message appears at the top of the Cluster Topology page:

```
You have more than one instance of the Application Server Control application
running in this cluster. This is not a recommended configuration and could lead
to unexpected problems. Please stop the additional instances of Application
Server Control or disable routing to these instances.
```

This problem can occur, for example, if you have installed two or more separate Oracle Application Server instances and you later combine them into a single cluster topology. Before joining the cluster, each Oracle Application Server instance has its own Application Server Control that is used to manage the instance. After joining the cluster, only one Application Server Control is required to manage the entire cluster.

To remedy this problem, do the following:

1. Navigate to the Cluster Topology page and click **Expand All** to see all the OC4J instances and the applications deployed to the instances.

2. Select any `ascontrol` applications that are not marked with the green, diamond icon that identifies the active ascontrol application.

3. Click **Stop**.

   This operation stops the selected non-active `ascontrol` applications. In addition, Application Server Control performs the following actions automatically whenever you stop a non-active ascontrol in a cluster:

   - To prevent the non-active ascontrol applications from starting the next time the host OC4J instance is restarted, Application Server Control modifies the `server.xml` file for the OC4J instance. Specifically, Application Server Control sets the `start` parameter for the application to `false`.

   - To prevent any Oracle HTTP Server instances from routing requests to the non-active ascontrol applications, Application Server Control modifies the default-web-site.xml file for the OC4J instance. Specifically, it sets the `ohs-routing` parameter for the ascontrol application to `false`.

   As a result, if you later need to start any non-active ascontrol applications—for example, if you later separate the Oracle Application Server instance from the cluster and want to manage it separately—you must reset these configuration settings to their original value.

   For more information, see Section A.6.4.

A.6.4 Identifying and Configuring a New Active Application Server Control

In some situations, you might want to identify a different instance of the `ascontrol` application as your active Application Server Control. For example, if you are not using a dedicated administration OC4J instance, you might reconfigure your environment so that the `ascontrol` application is running in its own OC4J instance on its own host.

Similarly, if you installed a new Oracle Application Server instance and did not identify it as an administration OC4J during the installation, you could end up with no `ascontrol` applications running. In this case, you must start one of the `ascontrol` applications and identify it as your active Application Server Control

In either of these scenarios, perform the following procedure:

1. For each instance of the ascontrol application that is deployed to your cluster, perform the following steps:

   a. Use a text editor to open the `server.xml` file for the OC4J instance:

      On UNIX systems:

      ORACLE_HOME/j2ee/*oc4j_instance_name*/config/server.xml

      On Windows systems:

      ORACLE_HOME\j2ee*oc4j_instance_name*\config\server.xml

      In the example, *oc4j_instance_name* is the name of the OC4J instance where the `ascontrol` application is deployed.

   b. For the active `ascontrol` application, make sure that the `start` argument is set to `true`. This ensures that the application will be started automatically whenever the OC4J instance is restarted.

      ```
      <application name="ascontrol"
                   path="../../oc4j_instance_name/applications/ascontrol.ear"
      ```

```
                          parent="system"
                          start="true" />
```

For all other instances of the `ascontrol` application, make sure that the start argument is set to `false`. This ensures that the selected instance of the application will *not* be started automatically when the OC4J instance is restarted.

c. Save your changes and close the `server.xml` file.

d. Open the `default-web-site.xml` file, which is stored in the same directory as the `server.xml` file.

e. For the active `ascontrol` application, change the `ohs-routing` attribute of the `web-app` element to `true`:

```
<web-app application="ascontrol"
         name="ascontrol"
         load-on-startup="true"
         root="/em"
         ohs-routing="true" />
```

For all other instances of the `ascontrol` application, make sure that the `ohs-routing` attribute is set to `false`. This prevents Oracle HTTP Server from routing HTTP requests to the non-active instances of the `ascontrol` application.

f. Save your changes and close the `default-web-site.xml` file.

2. After you have configured the `server.xml` and `default-web-site.xml` files for each OC4J instance in the cluster, perform the following steps:

a. From the Cluster Topology page, stop and restart all of the OC4J instances, except the instance that is used to deploy the currently active `ascontrol` application.

Note that the currently active `ascontrol` application is identified by the green diamond icon.

b. After you restart the other OC4J instances, restart the administration OC4J instance that hosts the currently active `ascontrol` application.

When you restart the currently active ascontrol application, you will be logged out of Application Server Control.

c. Log in to Application Server Control using the URL for the new active `ascontrol` application.

For example:

```
http://new_admin_oc4j_hostname:port/em
```

A.6.5 Accessing the Administration OC4J Directly Through HTTP

One of the advantages of using a dedicated administration OC4J instance is that you can easily configure the administration OC4J to use its own HTTP listener.

For example, if your administration OC4J is currently accessed by using Oracle HTTP Server, you can reconfigure the instance to use the built-in OC4J HTTP listener, which means you can bypass Oracle HTTP Server and access the administration OC4J directly using the HTTP protocol.

Oracle Application Server administrators can then access Application Server Control using a separate URL and port number. Meanwhile, the applications you deploy on

the other OC4J instances can still be accessed by your application users by using Oracle HTTP Server.

To further protect access to Application Server Control, the administration OC4J can reside on a separate host, but still manage the components of the cluster topology. Figure A–3 shows such a configuration.

Figure A–3   Managing the Cluster Topology Using the Administration OC4J HTTP Listener
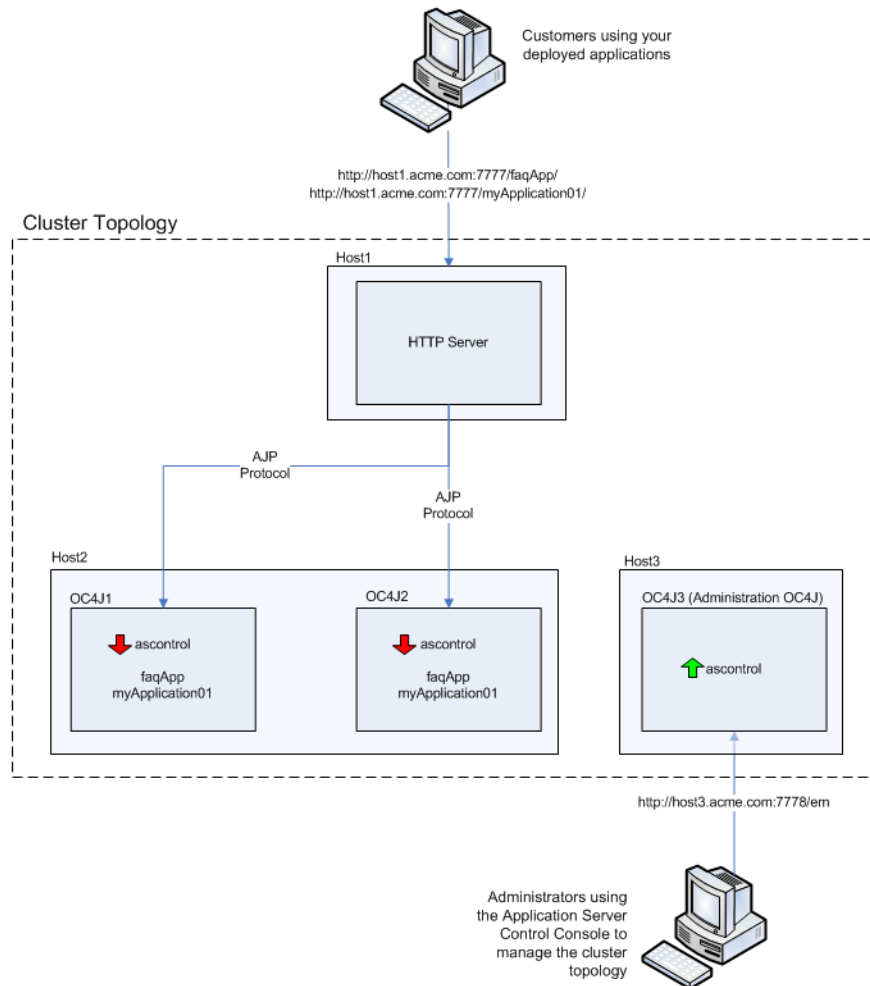
To configure an OC4J instance to use its own HTTP URL, instead of a front-end Oracle HTTP Server, change the protocol for the OC4J instance from AJP to HTTP and specify an HTTP port, as follows:

1. From the Cluster Topology page, click the name of the administration OC4J instance.

2. From the OC4J Home page, click **Administration**.

3. Click the task icon in the **Server Properties** row of the table on the OC4J Administration page.

4. In the **default-web-site** row of the Web sites table, select **http** from the **Protocol** drop-down menu and enter a port number or port range in the **Port** field.

If the administration OC4J instance resides on the same host as Oracle HTTP Server, be sure to use a port or port range that is different from the port used by Oracle HTTP Server.

5. Click **Apply** to apply the changes.

6. Navigate to the Cluster Topology page and restart the administration OC4J instance.

After you restart the administration OC4J, you can then access Application Server Control using the following URL:

```
http://admin_oc4j_hostname:port/em/
```

A.6.6 Publishing Application Server Control to a Separate Web Site in the Same OC4J Instance

If you do not use a dedicated administration OC4J, you can still configure the OC4J instance so that administrators bypass Oracle HTTP Server and access Application Server Control directly by using a separate HTTP port.

In this scenario, you can configure an additional Web site for the OC4J instance, migrate the existing Application Server Control bindings to the Web site, and configure OPMN to be aware of the Web site. The existing Web site (usually the `default-web-site`) is then still available for hosting your deployed applications, but Oracle Application Server administrators access Application Server Control using the new Web site and port.

Perform these steps to migrate Application Server Control to its own OC4J Web site:

1. Copy the `ORACLE_HOME`/j2ee/home/config/default-web-site.xml file to `ORACLE_HOME`/j2ee/home/config/ascontrol-web-site.xml (or a file name of your choice).

2. Edit the `ascontrol-web-site.xml` file to remove any existing web application bindings. Leave only the `<default-web-app>` entry and the `<web-app>` entry for Application Server Control application, shown in bold in the example. This maps the `ascontrol` application to the root context `/em` for the Web site. Ensure that the `<web-site>` element specifies `protocol="http"`, `port="1810"`, and that `display-name` is a unique name.

```
<?xml version = '1.0' standalone = 'yes'?>
<web-site
    protocol="http"
    port="1810
    display-name="OC4J 10g (10.1.3) ASControl Web Site"

    <default-web-app application="default" name="defaultWebApp" root="/j2ee" />
    <web-app application="ascontrol" name="ascontrol" root="/em" />

    <!-- Access Log, where requests are logged to -->
    <access-log path="../log/default-web-access.log"/>
    <!-- Uncomment this if you want to use ODL logging capabilities
    <odl-access-log path="../log/default-web-access" max-file-size="1000"
max-directory-size="10000"/>
    -->
    <web-app application="bc4j" name="webapp" root="/webapp"
load-on-startup="false"/>
</web-site>
```

3. Change the `access-log path` to specify a unique log file for the ascontrol Web site.

4. Edit *ORACLE_HOME*/j2ee/home/config/server.xml to add a new `<web-site>` element that specifies the `ascontrol-web-site.xml` file, as shown in bold in the following example:

```
<application-server ...>
...
        <web-site default="true" path="./default-web-site.xml" />
        <web-site default="false" path="./ascontrol-web-site.xml" />...
</application-server>
```

5. Edit *ORACLE_HOME*/j2ee/home/config/default-web-site.xml to remove or comment out the `web-app` binding for the `ascontrol` application.

```
<web-site
   protocol="http"
   port="1810"
   display-name="OC4J 10g (10.1.3) ASControl Web Site"
...
<!--
   <web-app application="ascontrol" name="ascontrol" root="/em" / -->

</web-site>
```

6. Update the OPMN configuration with the additional `ascontrol` Web site so that OPMN is aware of the port settings of the `ascontrol` Web site. Issue this command from *ORACLE_HOME*/opmn/bin:

```
opmnctl config port update ias-component=OC4J process-type=home
 portid=ascontrol-web-site protocol="http" range=1810-1820
```

7. Restart the server by issuing these commands in *ORACLE_HOME*/opmn/bin:

```
opmnctl stopall
opmnctl startall
```

Application Server Control is now accessible at `AppHost1:1810/em`, and is isolated from Oracle HTTP Server. However, the default application and other applications (deployed as children to the default application) will still use Oracle HTTP Server.

B

Oracle Application Server Command-Line Tools

Table B–1 summarizes the command-line tools available in Oracle Application Server, with descriptions and pointers to more information.

Table B–1   Oracle Application Server Command-Line Tools

Command	Path from Oracle Home	Description
bkp_restore	UNIX: backup_restore/bkp_restore.sh Windows: backup_restore\bkp_restore.bat	Back up and restore Oracle Application Server instances. **See:** Chapter 16
chgiphost	UNIX: chgip/scripts/chgiphost.sh Windows: chgip\scripts\chgiphost.bat	Change the hostname, domain name, or IP address of a middle-tier instance, Infrastructure, or Identity Management installation. **See:** Section 7.2.1
clone	UNIX: clone/bin/clone.pl Windows: clone\bin\clone.pl	Clone Oracle Application Server instances. **See:** Chapter 9
createASinstance	UNIX: confighomes/bin Windows: confighomes\bin	Create additional Application Server instances. **See:** Section 6.1
createinstance	UNIX: bin/createinstance.sh Windows: bin\createinstance.bat	Create new OC4J instances. **See:** Section 6.2
dmstool	UNIX: bin/dmstool Windows: bin\dmstool.bat	View performance metrics and set reporting intervals. **See:** *Oracle Application Server Performance Guide*
jazn.jar	UNIX: j2ee/home/jazn.jar Windows: j2ee\home\jazn.jar	Manage both XML-based and LDAP-based JAAS data. **See:** *Oracle Containers for J2EE Security Guide*
ojspc	UNIX: bin/ojspc Windows: bin\ojspc.bat	JSP back precompiler. **See:** *Oracle Containers for J2EE Support for JavaServer Pages Developer's Guide*
opmnassociate	UNIX: bin/opmnassociate Windows: bin\opmnassociate.cmd	Add an OC4J instance to a cluster. **See:** *Oracle Process Manager and Notification Server Administrator's Guide*
opmnctl	UNIX: opmn/bin/opmnctl Windows: opmn\bin\opmnctl.exe	Start, stop, and get status on OPMN-managed processes. **See:** *Oracle Process Manager and Notification Server Administrator's Guide*
orapki	UNIX: bin/orapki.sh Windows: bin\orapki.bat	Manage Certificate Revocation Lists (CRLs), manage wallets. **See:** Section 11.2.1

Table B–1   (Cont.) Oracle Application Server Command-Line Tools

Command	Path from Oracle Home	Description
prepare_clone	UNIX: clone/bin/prepare_clone.pl Windows: clone\bin\prepare_clone.pl	Prepare an Oracle Application Server instance for cloning. **See:** Chapter 9
removeinstance	UNIX: bin/removeinstance.sh Windows: bin\removeinstance.bat	Remove OC4J instances. **See:** Section 6.2

C

URLs for Components

Table C–1 shows the URLs and login IDs to use to access components after installation.

The URLs in the table are shown with the default ports. The components in your environment might use different ports. To determine the port numbers for components, use the following command.

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

Table C–1   URLs for Components

Component	URL (with Default Port Number)	Entry in opmnctl status Output	Login and Password
Welcome Page	UNIX: http://*host*:7777 Windows: http://*host*:80	HTTP_Server	Not Applicable
Oracle HTTP Server	UNIX: http://*host*:7777 Windows: http://*host*:80	HTTP_Server	Not Applicable
Application Server Control	UNIX: http://*host*:7777/em Windows: http://*host*:80/em	HTTP_Server	`oc4jadmin` Use the `oc4jadmin` password you supplied during installation.
Oracle Content DB	UNIX: http://*host*:7777/em Windows: http://*host*:80/em	HTTP_Server	`oc4jadmin` Use the `oc4jadmin` password you supplied during installation.
Rich Text Portlet	UNIX: http://*host*:7777/richtextportlet/portlets/wsrp2?WSDL Windows: http://*host*:80/richtextportlet/portlets/wsrp2?WSDL	HTTP_Server	Not Applicable

Table C–1   (Cont.) URLs for Components

Component	URL (with Default Port Number)	Entry in opmnctl status Output	Login and Password
OmniPortlet	UNIX: http://*host*:7777/portalTools/omniPortlet/providers/omniPortlet Windows: http://*host*:80/portalTools/omniPortlet/providers/omniPortlet	HTTP_Server	Not Applicable
Web Clipping	UNIX: http://*host*:7777/portalTools/webClipping/providers/webClipping Windows: http://*host*:80/portalTools/webClipping/providers/webClipping	HTTP_Server	Not Applicable

D

Oracle Application Server Port Numbers

This appendix provides information about Oracle Application Server port numbers.

It contains the following topics:

- Port Numbers and How They Are Assigned

  This section lists the allotted port range, the default port number, information about when the port number is assigned and where to find information about changing the port number.

- Port Numbers (Sorted by Number)

  This section provides a table that lists all allotted port ranges, sorted by number. It is useful for determining if a particular port number is used by Oracle Application Server.

- Ports to Open in Firewalls

  This section lists ports that must be opened if Oracle Application Server is installed behind a firewall.

D.1 Port Numbers and How They Are Assigned

Most port numbers are assigned by Oracle Application Server during installation. Oracle Application Server chooses a free port from the allotted port range.

This section provides the following information for each Oracle Application Server service that uses a port:

- **Component or Service:** The name of the component and service and information about when the port number is assigned and where to find information about changing the port number, if it can be changed.

- **Allotted Port Range:** The set of port numbers Oracle Application Server attempts to use when assigning a port.

- **Default Port Number:** The first port number Oracle Application Server attempts to assign to a service. It is usually the lowest number in the allotted port range.

- **Protocol:** Protocol used.

The ports are sorted into the following categories:

- OC4J, OPMN, and Oracle HTTP Server Ports

- Oracle WebCenter Framework and Oracle Content DB

- Port Numbers for Other Components

D.1.1 OC4J, OPMN, and Oracle HTTP Server Ports

Table D–1 lists many of the ports in a 10*g* Release 3 (10.1.3.4) installation. Unless otherwise noted in the Component or Service column of the table:

- The port number is assigned during installation.

- If the port number is assigned during installation, you can override the port number during installation by using the `staticports.ini` file.

  For most ports, you can override the default port assignment during installation by specifying a port number in `staticports.ini`. You create a template called `staticports.ini` with the port numbers you would like to use, and launch Oracle Universal Installer with special options.

  > **See Also:** *Oracle Application Server Installation Guide* for information on how to use `staticports.ini`

- You can change the port number after installation.

Table D–1   OC4J, OPMN, and Oracle HTTP Server Ports

Component or Service	Allotted Port Range	Default Port Number	Protocol
Oracle HTTP Server			
Listen Port See Section 4.3.3, "Changing the Oracle HTTP Server Listen Ports", or, if this is an Oracle WebCenter Framework only installation, see Section 4.3.1, "Changing the OC4J Listener Port" to change the port number.	80, 7777 - 7877, 8888	7777[1]	HTTP
Port See Section 4.3.3, "Changing the Oracle HTTP Server Listen Ports" to change the port number.	80, 7777 - 7877, 8888	7777[1]	HTTP
Listen (SSL) port This port is not used unless you enable SSL after installation. Refer to *Oracle HTTP Server Administrator's Guide*. See Section 4.3.3, "Changing the Oracle HTTP Server Listen Ports" to change the port number.	4443	4443	HTTPS
SSL Port This port is not used unless you enable SSL after installation. Refer to *Oracle HTTP Server Administrator's Guide*. See Section 4.3.3, "Changing the Oracle HTTP Server Listen Ports" to change the port number.	443, 4443	4443	HTTPS
Diagnostic See Section 4.3.4, "Changing the Oracle HTTP Server Diagnostic Port" to change the port number.	7200 - 7299	7200	TCP
OC4J			
AJP You cannot override this port number during installation. See Section 4.3.2, "Changing Other OC4J Ports" to change the port number.	12501 - 12600	12501	TCP

Table D–1   (Cont.) OC4J, OPMN, and Oracle HTTP Server Ports

Component or Service	Allotted Port Range	Default Port Number	Protocol
Default Web Site, with HTTP protocol	8888-8987	8888	HTTP
You cannot override this port number during installation.			
See Section 4.3.1, "Changing the OC4J Listener Port" to change the port number.			
JMS	12601 - 12700	12601	TCP
You cannot override this port number during installation.			
See Section 4.3.2, "Changing Other OC4J Ports" to change the port number.			
RMI	12401 - 12500	12401	TCP
You cannot override this port number during installation.			
See Section 4.3.2, "Changing Other OC4J Ports" to change the port number.			
RMIS	12701 - 12800	12701	TCP
You cannot override this port number during installation.			
See Section 4.3.2, "Changing Other OC4J Ports" to change the port number.			
IIOP	13301 - 13400	13301	TCP
Port is assigned after installation, when you configure IIOP. Refer to *Oracle Containers for J2EE Services Guide*.			
See Section 4.3.2, "Changing Other OC4J Ports" to change the port number.			
IIOPS1 (Server only)	13401 - 13500	13401	TCP
Port is assigned after installation, when you configure IIOPS1.			
See Section 4.3.2, "Changing Other OC4J Ports" to change the port number.			
IIOPS2 (Server and client)	13501 - 13600	13501	TCP
Port is assigned after installation, when you configure IIOPS2.			
See Section 4.3.2, "Changing Other OC4J Ports" to change the port number.			
OPMN			
ONS Local	6100 - 6199	6100	HTTP/TCP
See Section 4.3.6, "Changing OPMN Ports (ONS Local, Request, and Remote)" to change the port number.			
ONS Remote	6200 - 6299	6200	HTTP/TCP
See Section 4.3.6, "Changing OPMN Ports (ONS Local, Request, and Remote)" to change the port number.			
ONS Request	6003 - 6099	6003	HTTP/TCP
See Section 4.3.6, "Changing OPMN Ports (ONS Local, Request, and Remote)" to change the port number.			

Table D–1   (Cont.) OC4J, OPMN, and Oracle HTTP Server Ports

Component or Service	Allotted Port Range	Default Port Number	Protocol
Miscellaneous Services			
Java Object Cache See Section 4.3.5, "Changing the Java Object Cache Port" to change the port number.	7000 - 7099	7000	TCP
Port Tunneling Port is assigned after installation, when you configure Port Tunneling. See Section 4.3.7, "Changing the Port Tunneling Port" to change the port number.	7501 - 7599	7501	TCP

[1] On Windows, the default port is 80.

D.1.2 Oracle WebCenter Framework and Oracle Content DB

Oracle WebCenter Framework components and Oracle Content DB use the following ports:

- For most installation types, Oracle WebCenter Framework components use the Oracle HTTP Server port. However, if you install only Oracle WebCenter Framework, it uses the OC4J_WebCenter HTTP port.

- Oracle Content DB uses the Oracle HTTP Server port.

See Section D.1.1 for a list of these ports.

D.1.3 Port Numbers for Other Components

Other port numbers may be used by components that are installed separately from Oracle Application Server. See that component's documentation for information about port numbers.

For example, for more information about OracleAS Infrastructure, Release 2 (10.1.2), see the Release 2 (10.1.2) *Oracle Application Server Administrator's Guide*.

D.2 Port Numbers (Sorted by Number)

Table D–2 lists Oracle Application Server ports numbers and services, sorted in ascending order by port number.

Table D–2   Port Numbers (Sorted by Number)

Port Number	Service
80	Oracle HTTP Server Listen and Oracle HTTP Server Port (Windows only)
443	Oracle HTTP Server Port (SSL)
4443	Oracle HTTP Server Listen (SSL) and Oracle HTTP Server Port (SSL)
6003 - 6099	OPMN ONS Request
6100 - 6199	OPMN ONS Local
6200 - 6299	OPMN ONS Remote
7000 - 7099	Java Object Cache
7200 - 7299	Oracle HTTP Server Diagnostic

Table D–2 (Cont.) Port Numbers (Sorted by Number)

Port Number	Service
7501 - 7599	Port Tunneling
7777 - 7877	Oracle HTTP Server Listen and Oracle HTTP Server port
7890 - 7895	Oracle Application Server Guard
8250 - 8350	Oracle HTTP Server Listen (SSL) and Oracle HTTP Server port (SSL)
8888-8987	OC4J HTTP listener port
12401 - 12500	OC4J RMI
12501 - 12600	OC4J AJP
12601 - 12700	OC4J JMS
12701 - 12800	OC4J RMIS
13301 - 13400	OC4J IIOP
13401 - 13500	OC4J IIOPS1 (Server only)
13501 - 13600	OC4J IIOPS2 (Server and client)

D.3 Ports to Open in Firewalls

If you plan to install Oracle Application Server behind a firewall, you must open certain ports in the firewall during installation and run time.

For a 10*g* Release 3 (10.1.3.4) middle-tier instance, you need access to Oracle Notification Server and AJP ports. You need to open the following ports used by these components in the firewall:

- The OPMN ONS remote port
- OC4J AJP port

E

Examples of Administrative Changes

This appendix provides examples of administrative changes that can be performed on an Oracle Application Server environment. It is a companion to Part V, "Backup and Recovery" in this book, and to the Disaster Recovery section in *Oracle Application Server High Availability Guide*.

It contains the following topics:

- How to Use This Appendix
- Examples of Administrative Changes (by Component)

E.1 How to Use This Appendix

Some administrative operations cause configuration changes to your Oracle Application Server environment. These are called **administrative changes**, and include deploying and undeploying applications, changing the topology, changing ports, creating and deleting users, and changing passwords. As an administrator, you should be aware when administrative changes occur, because you may need to back up your environment or perform some synchronization procedures.

This appendix provides examples of administrative changes, listed by component. You can use this as a guide for performing the following procedures:

- Backup and Recovery

  Oracle recommends you perform a backup after each administrative change to your environment. You can use this appendix to determine the types of administrative changes that require you to back up your environment.

  > **See Also:** Part V, "Backup and Recovery"

- Disaster Recovery Synchronization Between the Primary and Standby Sites

  When you implement Disaster Recovery, you must update standby sites when you make an administrative change to your environment. You can use this appendix to determine the types of administrative changes that require you to update your standby sites.

  > **See Also:** *Oracle Application Server High Availability Guide*

E.2 Examples of Administrative Changes (by Component)

Table E–1 provides examples of administrative changes, by component. Consult your component documentation to learn more about these operations.

Table E–1   Examples of Administrative Changes

Component	Examples of Administrative Changes
Dynamic Monitoring Service (DMS)	DMS administrative and configuration operations performed using Application Server Control
	Manual edits to DMS configuration files, such as `dms.conf`
Oracle Containers for J2EE (OC4J)	OC4J administrative and configuration operations, such as deploying and undeploying applications, and creating OC4J instances, including those performed using Application Server Control
	Manual edits to OC4J configuration files
Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider (JAZN)	JAZN administrative and configuration operations performed using Application Server Control
	JAZN administrative and configuration operations performed using the `admintool` utility, such as adding and removing users, and changing roles, permissions, privileges, and passwords
Oracle Enterprise Manager 10*g* Application Server Control Console	Application server-wide or component-specific administrative and configuration operations performed using Application Server Control, such as changing the `oc4jadmin` password, deploying and undeploying applications, and operations that result in configuration file changes
Oracle HTTP Server	Oracle HTTP Server administrative and configuration operations, including those performed using Application Server Control, such as modifying the number of VMs and creating virtual hosts
	Manual edits to Oracle HTTP Server configuration files
Oracle Process Manager and Notification Server (OPMN)	OPMN administrative and configuration operations, including those performed using Application Server Control
	Manual edits to OPMN configuration files, such as `opmn.xml`

F

Supplementary Procedures for Configuring LDAP-Based Replicas

This appendix contains auxiliary procedures that are referred to in Section 8.3, "Moving 10.1.4 or 10.1.2 Identity Management to a New Host".

It contains the following topics:

- About LDAP-Based Replicas
- Installing and Setting Up an LDAP-Based Replica

F.1 About LDAP-Based Replicas

This section describes how to install and configure an LDAP-based replica. It contains the following topics:

- What Is an LDAP-Based Replica?
- How Is the LDAP-Based Replica Used for Changing Infrastructure Services?

F.1.1 What Is an LDAP-Based Replica?

Oracle Internet Directory replication is the process of copying and maintaining the same data (or naming context) on multiple directory servers. Simply put, replication is a means of having two identical directories that contain the same information. One directory is called the master (or supplier). This directory contains the master copy of the naming context. The other directory is called the replica (or consumer). The master supplies replication updates to the replica, which keeps the master and replica in sync.

There are different types of replicas. This procedure uses an LDAP-based replica, which means the protocol for transferring data between the master and the replica is LDAP.

> **See Also:** *Oracle Internet Directory Administrator's Guide* for more information on directory replication and LDAP-based replicas

For the purposes of this procedure, the master and replica directories are part of a larger environment that includes the Identity Management installations that contain the directories, and the Metadata Repositories that support them. This is called the LDAP-based replica environment, and it contains the following:

- **Master**—The Identity Management installation containing the Oracle Internet Directory that holds the master copy of the naming context. It supplies replication updates to the replica.

- **Master Repository**—The Metadata Repository that the master uses to store its Identity Management schemas.

- **Replica**—The Identity Management installation containing the replicated Oracle Internet Directory.

- **Replica Repository**—The Metadata Repository that the replica uses to store its Identity Management schemas.

Figure F–1 illustrates the LDAP-based replica environment.

Figure F–1   LDAP-Based Replica Environment
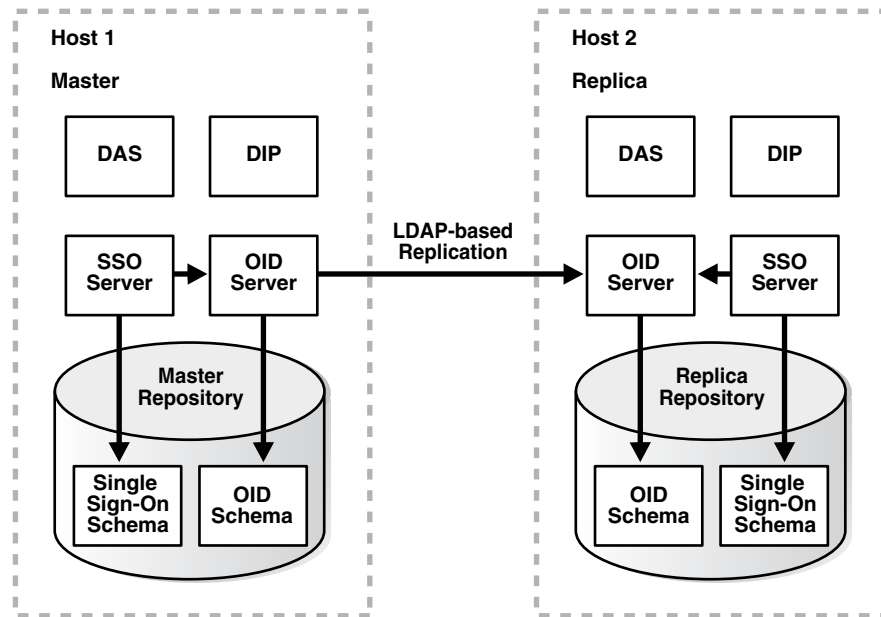
F.1.2 How Is the LDAP-Based Replica Used for Changing Infrastructure Services?

Typically, an LDAP-based replica is used to provide high availability and improved performance for directory users.

For the purposes of changing Infrastructure services in Section 8.3, the LDAP-based replica is created as a way of moving Identity Management from one host to another. The Master is the original Identity Management installation, and the Replica is the new Identity Management installation. In this case, replication is used to create an identical copy of the original Identity Management on a new host. You can then change your middle tiers from the old Identity Management (Master) to the new Identity Management (Replica) and discard the Master.

F.2 Installing and Setting Up an LDAP-Based Replica

This section describes how to install and set up an LDAP-based replica environment.

F.2.1 Things to Know Before You Start

You should be aware of these important items before you start the procedure:

- This procedure uses a single Infrastructure Oracle home that contains Identity Management and the Metadata Repository. However, you can split the Infrastructure installation so that Identity Management is in one Oracle home and

the Metadata Repository is in another Oracle home. You can also distribute the Identity Management components (OracleAS Single Sign-On, Oracle Internet Directory, Delegated Administration Services, Directory Integration and Provisioning) across different hosts. If you do this, perform the operations on each component in their respective Oracle homes.

- The replica always uses port 389 for the non-SSL Oracle Internet Directory port, and 636 for the SSL Oracle Internet Directory port, regardless of what is reported by Oracle Universal Installer. Make sure no other processes are using ports 389 and 636 on the replica host before you start the procedure.

- Make sure you use the `ldapsearch` and `ldapmodify` commands that are in `ORACLE_HOME`/bin. (Some operating systems ship their own version of these commands—do not use those.)

- These procedures use the `remtool` and `oidpasswd` commands. The messages returned by these commands are in UTF-8 encoding and are unreadable in most non-English environments. To work around this, set the `NLS_LANG` environment variable to `american_america.`*character_set* before running these commands. Most character sets (for example, `US7ASCII`) will work.

> **See Also:** *Oracle Application Server Globalization Guide*

- Make sure the `ORACLE_HOME` and `ORACLE_SID` environment variables are set. This applies to all platforms.

F.2.2 Procedure

This section contains the procedure for setting up an LDAP-based replica. It contains the following tasks:

- Task 1: Obtain the Master and Master Repository
- Task 2: Install Middle-Tier Instances (Optional)
- Task 3: Install and Configure the Replica

Task 1: Obtain the Master and Master Repository

Most likely, you already have your Master and Master Repository.

If you are following the procedure in Section 8.3, "Moving 10.1.4 or 10.1.2 Identity Management to a New Host", the Master and Master Repository are the installations you would like to move to a new host, and the LDAP-based replica will be the relocated installations.

If you are starting from the beginning, you can install a Master and Master Repository as follows:

1. Install Oracle Application Server using Oracle Universal Installer.

2. Choose the Infrastructure Installation.

3. Choose to install Identity Management and OracleAS Metadata Repository.

4. Choose to configure the following components: Oracle Internet Directory, OracleAS Single Sign-On, Delegated Administration Services, and Directory Integration and Provisioning.

Task 2: Install Middle-Tier Instances (Optional)

Most likely, you already have middle-tier instances using the Master for Identity Management services. This is fine, and, if desired, you can install and configure

additional instances to use the Master now, or at the end of this procedure after you have configured the Replica, or both.

These middle-tier instances can use the Master Repository for their product metadata, or they can use a different repository.

Task 3: Install and Configure the Replica

You can install and configure the Replica using Oracle Universal Installer. Be sure to install the Replica on a different host than the Master.

> **See Also:** *Oracle Application Server Installation Guide* for information on installing an Oracle Internet Directory replica

When the installation has finished, replication is configured and all components are started. You can return to the main procedure from where you started (Section 8.3, "Moving 10.1.4 or 10.1.2 Identity Management to a New Host").

G

Viewing Oracle Application Server Release Numbers

This appendix describes how to view Oracle Application Server release numbers.

It contains the following topics:

- Release Number Format
- Viewing Oracle Application Server Installation Release Numbers
- Viewing Component Release Numbers
- Using the OPatch Utility

> **Note:** Oracle recommends you keep a log of all interim patches applied to your Oracle Application Server installations.

G.1 Release Number Format

To understand the release level nomenclature used by Oracle, examine the example of an Oracle Application Server release number shown in Figure G–1.

Figure G–1 Example of an Oracle Application Server Release Number
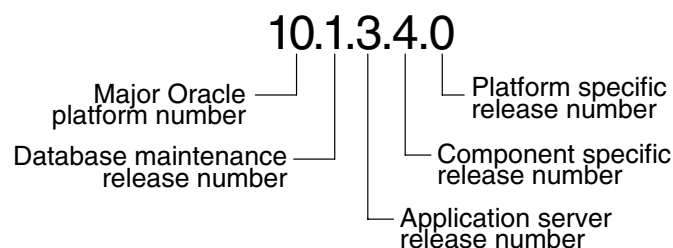
In Figure G–1, each digit is labeled:

- Major Oracle platform number

  This is the most general identifier. It represents a major new edition (or version) of an application, such as Oracle database server or Oracle Application Server, and indicates that the release contains significant new functionality.

- Database maintenance release number

  This digit represents a maintenance release level. Some new features may also be included.

- Application Server release number

    This digit reflects the release level of Oracle Application Server.

- Component-specific release number

    This digit identifies a release level specific to a component. Different components can have different numbers in this position depending upon, for example, component patch sets or interim releases.

- Platform-specific release number

    This digit identifies a platform-specific release.

G.2 Viewing Oracle Application Server Installation Release Numbers

All Oracle Application Server 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 Application Server installation using Oracle Universal Installer, as follows:

1. Launch Oracle Universal Installer:

    ```
    (UNIX) ORACLE_HOME/oui/bin/runInstaller.sh
    (Windows) ORACLE_HOME\oui\bin\setup.exe
    ```

2. Click **Installed Products** to open the Inventory Page.

3. In the Inventory Page, expand **Oracle Homes**. You will see entries for all installations on your host.

4. Expand the Oracle home entry for the installation you are interested in.

5. You will see an entry with the release number for your original installation, followed by entries for any patch sets that have been applied.

G.3 Viewing Component Release Numbers

All Oracle Application Server components have a release number and many contain services that have release numbers. These numbers *may* be updated when you apply a patch set release or upgrade the installation.

You can view the release number of components and their services in the following ways:

- On the File System
- Using Oracle Universal Installer

On the File System

You can view component release numbers as follows on UNIX:

```
cd ORACLE_HOME/inventory
ls -d Components*/*/*
```

Using Oracle Universal Installer

If you installed Oracle Application Server using Oracle Universal Installer, you can view component release numbers as follows:

1. Launch Oracle Universal Installer:

    ```
    (UNIX) ORACLE_HOME/oui/bin/runInstaller.sh
    ```

(Windows) *ORACLE_HOME*\oui\bin\setup.exe

2. Click **Installed Products** to open the Inventory Page.

3. In the Inventory Page, expand **Oracle Homes**. You will see entries for all installations on your host.

4. Expand the Oracle home entry for the installation you are interested in.

5. You will see an entry with the release number for your original installation, followed by entries for any patch sets that have been applied.

6. Expand the initial entry to view the component release numbers at installation time. If you have subsequent patch set entries, expand them to see the component release numbers updated for each patch set.

G.4 Using the OPatch Utility

The OPatch utility is a tool that allows the application and rollback of interim patches to Oracle products, such as Oracle Application Server. For the latest information about the opatch utility, and to check for updates, refer to Oracle MetaLink at

http://www.oracle.com/support/metalink/index.html

G.4.1 Requirements

The OPatch utility has the following requirements:

- Perl environment, included with Oracle Application Server or downloaded with a patch set.

- The Oracle home environment variable (ORACLE_HOME) must point to a valid Oracle home directory and match the value used during installation of the Oracle home directory.

- If the -invPtrLoc command-line argument was used during installation, then it must be used when using the OPatch utility. Oracle recommends the use of the default central inventory for a platform.

- The jar, java, ar, cp, and make commands must be available in the PATH statement. The commands are not available for all platforms.

- The library path must be set correctly for Oracle Real Application Clusters environments. Refer to the FAQ document in the opatch/doc directory for additional information.

> **See Also:** For the latest information about the OPatch utility, and to check for updates, refer to Oracle*MetaLink* at
>
> http://www.oracle.com/support/metalink/index.html

G.4.2 Running the OPatch Utility

The OPatch utility is located in the ORACLE_HOME/OPatch directory. The following shows the syntax for the OPatch utility:

path_to_opatch/opatch *option -command_line_arguments*

In the preceding example:

- *option*—the OPatch option. Values are described in Table G–1.

- *command_line_arguments*—the command-line arguments for each option. Values are described in the following sections.

Table G–1   Options for the OPatch Utility

Option	Description
apply	Installs an interim patch. See Section G.4.2.1.
lsinventory	Lists what is currently installed on the system. See Section G.4.2.2.
query	Queries a given patch for specific details. See Section G.4.2.3.
rollback	Removes an interim patch. See Section G.4.2.4.
version	Prints the current version of the patch tool. See Section G.4.2.5.

To view additional information for any option, use the following command:

```
path_to_OPatch/opatch option -help
```

If using Perl, then use the following command:

```
perl opatch.pl option -help
```

G.4.2.1 apply Option

The `apply` option applies an interim patch to a specified Oracle home. The ORACLE_HOME environment variable must be set to the Oracle home to be patched. The following syntax is used for this option:

```
path_to_opatch/opatch apply patch_location [-delay value] [-force] \
[-invPtrLoc path] [-jdk location] [-jre location] [-local] \
[-minimize_downtime] [-no_bug_superset] [-no_inventory] \
[-oh Oracle_home_location] \
[-post options_to _be_passed_into_post [-opatch_post_end]]\
[-pre options_to_be_passed_into_pre [-opatch_pre_end]] \
[-retry value] [-silent] [-verbose]
```

The following table lists the command-line arguments for the `apply` option:

Argument	Description
delay	Specifies how many seconds to wait before attempting to lock the inventory in the case of a previous failure.
force	Removes conflicting patches from the system. If a conflict exists which prevents the patch from being applied, then the -force argument can be used to apply the patch.
invPtrLoc	Specifies the location of the oraInst.loc file. This argument is needed when the -invPtrLoc argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
jdk	Specifies the location of a particular JDK (jar) to use instead of the default location under the Oracle home directory.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the Oracle home directory.

Argument	Description
local	Specifies that the OPatch utility patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes.
	This argument can be used on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shutdown before patching, then this argument can be used for non-rolling patches.
minimize_downtime	Specifies the order of nodes to be patched by the OPatch utility.
	This argument only applies to Oracle Real Application Clusters environments. It cannot be used with the -local argument or a rolling patch.
no_bug_superset	Specifies that the utility return an error if the current patch bugs-to-fix is a superset or the same as an installed patch bugs-fixed in the Oracle home directory.
no_inventory	Bypasses the inventory for reading and updates. This argument cannot be used with the -local argument. This argument puts the installation into an unsupported state.
oh	Specifies the Oracle home directory to use instead of the default.
opatch_post_end	Marks the end of the post options. This argument is used with the post argument. If this argument is not used, then everything after post is passed into post.
opatch_pre_end	Marks the end of the pre options. This argument is used with the pre argument. If this argument is not used, then everything after pre is passed into pre.
post	Specifies the parameters to be passed inside the post script besides the standard parameters.
pre	Specifies the parameters to be passed inside the pre script besides the standard parameters.
retry	Specifies how many times the OPatch utility should try when there is an inventory lock failure.
patch_location	Specifies the directory of the interim patch. This should be a directory with the same name as the patch.
silent	Suppresses user interaction, and defaults any answers to "yes."
verbose	Prints output to the screen as well as to the log file.

> **Note:** If a patch consists of SQL changes, then they are only staged. Follow the instructions included with the patch to apply the patch manually on the affected instances. For some products, the SQL application may be implemented as a post-staging action by the tool. These patches cannot be rolled back.

G.4.2.2 lsinventory Option

The lsinventory option reports what has been installed on the system for a particular Oracle home directory, or for all installations. The following syntax is used for this option:

```
path_to_opatch/opatch lsinventory [-all] [-detail] [-invPrtLoc path] \
[-jre location] [-oh Oracle_home_location]
```

The following table lists the command-line arguments for the `lsinventory` option:

Argument	Description
all	Reports the name and installation directory for each found Oracle home directory.
detail	Reports the installed products and other details. This argument cannot be used with the `-all` argument.
invPtrLoc	Specifies the location of the `oraInst.loc` file. This argument is needed when the `invPtrLoc` command-line argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the Oracle home directory.
oh	Specifies the Oracle home directory to use instead of the default directory.

The following is a sample output of `opatch lsinventory -detail`:

```
ORACLE_HOME      LOCATION
-----------      --------
Home1            /private/phi_local/OraHome1
   There is no Interim Patch
Home2            /private/phi_local/OraHome2
   There is no Interim Patch
Home3            /private/phi_local/OraHome6
   Installed Patch List:
    ====================
   1) Patch 20 applied on Mon Jul 11 15:53:51 PDT 2006
      [ Base Bug(s): 21  ]
   2) Patch 80 applied on Fri Jul 01 16:15:52 PDT 2006
      [ Base Bug(s): 80 81  ]
```

G.4.2.3 query Option

The `query` option queries a specific patch for specific details. It provides information about the patch and the system being patched. The following syntax is used for this option:

```
path_to_opatch/opatch query [-all] [-get_base_bug] [-get_component] \
[-invPtrLoc path] [-get_date] [-get_os] [-get_system_change] [-is_rolling] \
```

The following table lists the command-line arguments for the `query` option:

Argument	Description
all	Retrieves all information about a patch. This is equivalent to setting all command-line arguments.
get_base_bug	Describes the base bugs fixed by a patch.
get_component	Describes the Oracle components, optional or required, for a patch.
get_date	Provides the build date of a patch.
get_os	Provides the operating system description supported by a patch.
get_system_change	Describes the changes that will be made to the system by a patch. This argument is not available.

Argument	Description
invPtrLoc	Specifies the location of the oraInst.loc file. This argument is needed when the invPtrLoc command-line argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
is_rolling	Specifies if the patch is a rolling patch for Oracle Real Application Clusters. The set of patches need not be applied to the whole cluster at the same time. The patches can be applied to a select set of nodes at a time.

G.4.2.4 rollback Option

The rollback option removes a specific interim patch from the appropriate Oracle home directory. The following syntax is used for this option:

```
path_to_opatch/opatch rollback -id patch_id -ph patch_directory \
[-delay value] [-invPtrLoc path] [-jdk location] [-jre location]\
[-local] [-oh Oracle_home_location] \
[-post options_to_be_passed_into_post [-opatch_post_end]] \
[-pre options_to_be_passed_into_pre [-opatch_pre_end]] [-retry value] \
[-silent] [-verbose]
```

The following table lists the command-line arguments for the rollback option:

Argument	Description
delay	Specifies how many seconds the OPatch utility should wait before attempting to lock inventory again, if the -retry argument is used with the apply option.
id	Indicates the patch to be rolled back. Use the -lsinventory option to display all patch identifiers. To successfully rollback a patch, the patch identifier must be supplied.
invPtrLoc	Specifies the location of the oraInst.loc file. This argument is needed when the -invPtrLoc command-line argument was used during installation. Oracle recommends the use of the default central inventory for a platform.
jdk	Specifies the location of a particular JDK (jar) to use instead of the default location under the Oracle home directory.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the Oracle home directory.
local	Specifies that the OPatch utility patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes.
	This argument can be used on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shutdown before patching, then this argument can be used for non-rolling patches.
oh	Specifies the Oracle home directory to use instead of the default directory.
opatch_post_end	Marks the end of the post options. This argument is used with the post argument. If this argument is not used, then everything after post is passed into post.
opatch_pre_end	Marks the end of the pre options. This argument is used with the pre argument. If this argument is not used, then everything after pre is passed into pre.

Argument	Description
ph	Specifies the valid patch directory area. The utility will use the command types found in the patch directory to identify which commands are used for the current operating system.
post	Specifies the parameters to be passed inside the `post` script besides the standard parameters.
pre	Specifies the parameters to be passed inside the `pre` script besides the standard parameters.
retry	Specifies how many times the OPatch utility should try in case of an inventory lock failure.
silent	Suppresses user interaction, and defaults any answers to "yes."
verbose	Prints output to the screen as well as to the log file.

G.4.2.5 version Option

The `version` option shows the current version number of the OPatch utility. The following syntax is used for this option:

path_to_opatch/opatch version

H

Troubleshooting Oracle Application Server

This appendix provides information on how to troubleshoot problems that you might encounter when using Oracle Application Server. It contains the following topics:

- Diagnosing Oracle Application Server Problems
- Common Problems and Solutions
- Troubleshooting Application Server Control
- Need More Help?

> **See Also:**
>
> - Chapter 14, "Troubleshooting SSL" for information about troubleshooting SSL.
>
> - Chapter 19, "Troubleshooting OracleAS Recovery Manager" for specific information about troubleshooting the OracleAS Recovery Manager.

H.1 Diagnosing Oracle Application Server Problems

Oracle Application Server components generate log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, access information on HTTP requests, and additional information. The log files can be used to identify and diagnose problems. See Chapter 5, "Managing Log Files" for more information about using and reading log files.

H.2 Common Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

- Application Performance Impacted by Garbage Collection Pauses
- Application Server Returns Connection Refused Errors
- Oracle HTTP Server Unable to Start Due to Port Conflict
- Machine Overloaded by Number of HTTPD Processes
- Oracle Application Server Process Does Not Start
- OPMN Start Up Consumes CPU Processing Capability
- Browser Displaying a Page Not Displayed Error
- Standby Site Not Synchronized

■ [Failure to Bring Up Standby Instances After Failover or Switchover](#)

H.2.1 Application Performance Impacted by Garbage Collection Pauses

Application performance slows or application is unresponsive.

See the section "Application Performance Impacted by Garbage Collection Pauses" in the *Oracle Containers for J2EE Configuration and Administration Guide* for information about the causes and solutions to this problem.

H.2.2 Application Server Returns Connection Refused Errors

In load conditions (for example, when the number of users concurrently connecting to the application server increases significantly in a short time), the server may respond with the following error message:

```
IOException in sending request - Connection refused
```

Problem

If the number of simultaneous users is increasing, the server may be utilizing the maximum Oracle HTTP Server child processes allowable to service requests.

Solution

You may need to increase the MaxClients directive for the Oracle HTTP Server. The MaxClients directive specifies a limit on the number of clients who can simultaneously connect.

Use one of the following methods to determine if this is the case:

■ Search the Oracle HTTP Server error log file for the following message:

```
server reached MaxClients setting, consider raising the MaxClients setting
```

By default, the error log file is located in:

```
(UNIX) ORACLE_HOME/Apache/Apache/logs/error_log
(Windows) ORACLE_HOME\Apache\Apache\logs\error_log
```

■ Interactively monitor child process activity using the metrics in Application Server Control. In particular, view the following in the HTTP_Server Home page:

– In the Status section, **Active Connections,** which shows the number of clients currently executing HTTP requests.

– In the Response and Load section, **Active Requests,** which shows the total number of active requests currently being processed.

■ Look at the information provided by mod_status. The mod_status module provides an HTML page that shows the current server statistics. Check to see if the all the processes are busy. (By default, mod_status is enabled for localhost access only.) For more information on mod_status, see:

http://httpd.apache.org/docs/mod/mod_status.html

In addition, consider increasing the maximum queue length for pending connections (the ListenBackLog directive) and consider the impact of persistent connections (the KeepAlive directive).

For more information about the Oracle HTTP Server directives and how to change their values, see the *Oracle HTTP Server Administrator's Guide*. For more information

about tuning Oracle HTTP Server processes, see the *Oracle Application Server Performance Guide*.

H.2.3 Oracle HTTP Server Unable to Start Due to Port Conflict

You can get the following error if Oracle HTTP Server is unable to start due to port conflict:

```
[crit] (98) Address already in use: make_sock: could not bind to port 7778
```

See the section "Oracle HTTP Server Unable to Start Due to Port Conflict" in the Troubleshooting Oracle HTTP Server appendix of the *Oracle HTTP Server Administrator's Guide* for information about the cause and solution to this problem.

H.2.4 Machine Overloaded by Number of HTTPD Processes

When there are too many httpd processes running on a machine, the response time plummets.

See the section "Machine Overloaded by Number of HTTPD Processes" in the Troubleshooting Oracle HTTP Server appendix of the *Oracle HTTP Server Administrator's Guide* for information about the cause and solution to this problem.

H.2.5 Oracle Application Server Process Does Not Start

You are not able to start an Oracle Application Server process using OPMN.

See Section A.1.1, "Oracle Application Server Process Does Not Start" in the *Oracle Process Manager and Notification Server Administrator's Guide* for information about the causes and solutions to this problem.

H.2.6 OPMN Start Up Consumes CPU Processing Capability

On some computers, when OPMN starts up, it consumes large amounts of CPU processing capability.

See Section A.1.11, "OPMN Start Up Consumes CPU Processing Capability" in the *Oracle Process Manager and Notification Server Administrator's Guide* for information about the causes and solutions to this problem.

H.2.7 Browser Displaying a Page Not Displayed Error

Browsers return an error saying that a page cannot be displayed.

If you use OracleAS Web Cache, Release 2 (10.1.2), as a reverse proxy, see the section "Browser Displaying a Page Not Displayed Error" in the *Oracle Application Server Web Cache Administrator's Guide* for the release for information about the causes and solutions to this problem.

H.2.8 Standby Site Not Synchronized

In the OracleAS Disaster Recovery standby site, you may find that the site's OracleAS Metadata Repository is not synchronized with the OracleAS Metadata Repository in the primary site.

See Section A.1.1, "Standby Site Not Synchronized" in the *Oracle Application Server High Availability Guide* for information about the causes and solutions to this problem.

H.2.9 Failure to Bring Up Standby Instances After Failover or Switchover

Standby instances are not started after a failover or switchover operation.

See Section A.1.2, "Failure to Bring Up Standby Instances After Failover or Switchover" in the *Oracle Application Server High Availability Guide* for information about the causes and solutions to this problem.

H.3 Troubleshooting Application Server Control

The following sections describe problems and issues when using Application Server Control:

- Resetting the Administrator (oc4jadmin) Password
- Deployment Performance in Internet Explorer 6.0 and Netscape Navigator 7.0
- Troubleshooting OC4J Out-of-Memory Errors
- "403 Forbidden - Directory browsing not allowed" Error When Testing a Web Module or Web Service
- Administrator Credentials Error When Attempting to Access the OC4J Home Page in a Cluster Topology

H.3.1 Resetting the Administrator (oc4jadmin) Password

To manage an instance of Oracle Application Server, you must log in to Application Server Control using the current Administrator (oc4jadmin) password.

Problem

If you forget or do not know the oc4jadmin password, then you cannot monitor or administer the application server or its components with Application Server Control.

Solution

Reset the oc4jadmin password using the following procedure while you are logged in as the user who installed the Oracle Application Server instance:

1. Stop OC4J and the Application Server Control.

    Enter the following command in the Oracle home of the application server instance:

    ```
    (UNIX) ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=OC4J
    (Windows) ORACLE_HOME\opmn\bin\opmnctl stopproc ias-component=OC4J
    ```

2. Locate and open the following file in a text editor:

    ```
    (UNIX)ORACLE_HOME/j2ee/home/config/system-jazn-data.xml
    (Windows)ORACLE_HOME\j2ee\home\config\system-jazn-data.xml
    ```

3. Locate the line that defines the credentials property for the oc4jadmin user.

    The following example shows the section of system-jazn-data.xml with the encrypted credentials entry in boldface type:

    ```
    <jazn-realm>
       <realm>
          <name>jazn.com</name>
          <users>
           .
           .
    ```

```
.
<user>
   <name>oc4jadmin</name>
   <display-name>OC4J Administrator</display-name>
   <description>OC4J Administrator</description>
   <credentials>{903}4L50lHJWIFGwLgHXTub7eYK9e0AnWLUH</credentials>
</user>
```

4. Replace the existing encrypted password with the new password.

   Be sure to prefix the password with an exclamation point (!). For example:

   ```
   <credentials>!mynewpassword123</credentials>
   ```

   The password for the `oc4jadmin` user should conform to following guidelines:

   - Must contain at least five characters, but not more than 30 characters.

   - Must begin with an alphabetic character. It cannot begin with a number, the underscore (_), the dollar sign ($), or the number sign (#).

   - At least one of the characters must be a number.

   - Can contain only the following characters; numbers, letters, and the following special characters: US dollar sign ($), number sign (#), or underscore (_).

   - Cannot contain any Oracle reserved words, such as VARCHAR.

     See Also: "The oc4jadmin User and Restrictions on its Password" in the *Oracle Application Server Installation Guide*

5. Delete cached password data by deleting the contents of the following directory:

   ```
   (UNIX)
   ORACLE_HOME/j2ee/oc4jinstance/persistence/ascontrol/ascontrol/securestore/
   (Windows)
   ORACLE_HOME\j2ee\oc4jinstance\persistence\ascontrol\ascontrol\securestore\
   ```

6. Start OC4J and Application Server Control.

   After the restart, Application Server Control will use your new Administrator (`oc4jadmin`) password, which will be stored in encrypted format within the `system-jazn-data.xml` file.

     See Also: Section A.1, "Starting and Stopping Application Server Control"

H.3.2 Deployment Performance in Internet Explorer 6.0 and Netscape Navigator 7.0

Problem

If you attempt to deploy an OC4J application while using Microsoft Internet Explorer 6.0 or Netscape 7.0, the file upload may take an extremely long time (for example, 10 minutes for a 45 MB `.ear` file as compared to 15 seconds with Netscape 7.1).

Solution

If you are using Netscape Navigator, upgrade to Netscape 7.1.

If you are using Internet Explorer, refer to the following Microsoft knowledge base article, which addresses this problem:

http://support.microsoft.com/default.aspx?scid=kb;en-us;329781

H.3.3 Troubleshooting OC4J Out-of-Memory Errors

Problem

Depending upon the size and number of applications you deploy to your OC4J instance, you might experience "out of memory" errors.

Solution

Adjust the Java Virtual Machine (JVM) heap size for your OC4J processes.

> **See Also:**
> - "Setting the JVM Heap Size for OC4J Processes" in the chapter "Optimizing J2EE Applications In OC4J" in the *Oracle Application Server Performance Guide*
> - "Administering OC4J Server Properties" in Application Server Control online help

H.3.4 "403 Forbidden - Directory browsing not allowed" Error When Testing a Web Module or Web Service

Problem

From the Application Server Control Console, you can test your Web modules to make sure they are working properly. However, in many cases, when you click the **Test Module** or **Test Service** button, the following error occurs in your Web browser:

- In Mozilla Firefox:

  ```
  403 Forbidden - Directory browsing not allowed
  ```

- In Microsoft Internet Explorer:

  ```
  You are not authorized to view this page.
  You might not have permission to view this directory or page using the
  credentials you supplied.
  ```

Solution

Application Server Control cannot always determine the complete URL for the Web module or Web service you are testing. Instead, Application Server Control attempts to assemble the URL based on information about the root context of the Web module or Web service.

To work around this problem, use the text field on the Test Web Module or Test Web service page to enter the complete URL to your application; then, click **Test Module** or **Test Service**.

H.3.5 Administrator Credentials Error When Attempting to Access the OC4J Home Page in a Cluster Topology

Problem

From the Cluster Topology page, you click the name of an OC4J instance and you are prompted to enter your administrator credentials. However, when you enter the proper administrator credentials, Application Server Control displays the following error:

```
Administrator credentials were saved but they cannot be used to make a
```

```
connection.  Enter new credentials or cancel.
```

Solution

Check to see if the Administration OC4J instance has been configured to use the secure Remote Method Invocation (RMIS) protocol. If a remote OC4J instance you are managing is not configured for RMIS, depending upon the configuration, you may not be able to access the OC4J Home page.

For information on enabling RMIS connections for Application Server Control, see Section A.3.

H.4 Need More Help?

You can find more solutions on Oracle*MetaLink*, `http://metalink.oracle.com`. If you do not find a solution for your problem, log a service request.

> **See Also:**   *Oracle Application Server Release Notes*, available on the Oracle Technology Network:
>
> `http://www.oracle.com/technology/documentation/index.html`

Glossary

access control

The ability of a system to grant or limit access to specific data for specific clients or groups of clients.

Access Control Lists (ACLs)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients, or groups of clients, or both.

administration OC4J instance

The OC4J instance used to deploy the active `ascontrol` application.

Advanced Encryption Standard

A cryptographic algorithm that has been approved by the National Institute of Standards and Technology as a replacement for DES. The AES standard is available in Federal Information Processing Standards Publication 197. The AES algorithm is a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits.

AES

See **Advanced Encryption Standard**.

attribute

An item of information that describes some aspect of an entry in an LDAP directory. An entry comprises a set of attributes, each of which belongs to an **object class**. Moreover, each attribute has both a *type*, which describes the kind of information in the attribute, and a *value,* which contains the actual data.

authentication

The process of verifying the identity of a user, device, or other entity in a computer system, often as a prerequisite to granting access to resources in a system. A recipient of an authenticated message can be certain of the message's origin (its sender). Authentication is presumed to preclude the possibility that another party has impersonated the sender.

authentication method

A security method that verifies the identity of a user, client, or server in distributed environments. Network authentication methods can also provide the benefit of **single sign-on (SSO)** for users. The following authentication methods are supported in Oracle Application Server:

■ **Kerberos**

- **Secure Sockets Layer (SSL)**

- **Windows native authentication**

authorization

The evaluation of security constraints to send a message or make a request. Authorization uses specific criteria to determine whether the request should be permitted. The criteria are **authentication** and restriction.

auto login wallet

An Oracle Wallet Manager feature that enables PKI- or password-based access to services without providing credentials at the time of access. This auto login access stays in effect until the auto login feature is disabled for that wallet. File system permissions provide the necessary security for auto login wallets. When auto login is enabled for a wallet, it is only available to the operating system user who has permission to read that wallet. Sometimes these are called "SSO wallets" because they provide single sign-on capability.

base

The root of a subtree search in an **LDAP**-compliant directory.

CA

See **certificate authority**.

certificate

An ITU X.509 Version 3 standard data structure that securely binds an identity to a public key.

A certificate is created when an entity's public key is signed by a trusted identity, a certificate authority. The certificate ensures that the entity's information is correct and that the public key actually belongs to that entity.

A certificate contains the entity's name, identifying information, and public key. It is also likely to contain a serial number, expiration date, and information about the rights, uses, and privileges associated with the certificate. Finally, it contains information about the certificate authority that issued it.

certificate authority

A trusted third party that certifies that other entities—users, databases, administrators, clients, servers—are who they say they are. When it certifies a user, the certificate authority first seeks verification that the user is not on the certificate revocation list (CRL), then verifies the user's identity and grants a certificate, signing it with the certificate authority's private key. The certificate authority has its own certificate and public key which it publishes. Servers and clients use these to verify signatures the certificate authority has made. A certificate authority might be an external company that offers certificate services, or an internal organization such as a corporate MIS department.

certificate chain

An ordered list of certificates containing an end-user or subscriber certificate and its certificate authority certificates.

certificate request

A request, which consists of three parts: certification request information, a signature algorithm identifier, and a digital signature on the certification request information.

The certification request information consists of the subject's distinguished name, public key, and an optional set of attributes. The attributes may provide additional information about the subject identity, such as postal address, or a challenge password by which the subject entity may later request certificate revocation. See **PKCS #10**.

certificate revocation lists

Signed data structures that contain a list of revoked **certificate**s. The authenticity and integrity of the certificate revocation list (CRL) is provided by a digital signature appended to it. Usually, the CRL signer is the same entity that signed the issued certificate.

Cipher Block Chaining (CBC)

An encryption method that protects against block replay attacks by making the encryption of a cipher block dependent on all blocks that precede it; it is designed to make unauthorized decryption incrementally more difficult. Oracle Advanced Security employs *outer* cipher block chaining because it is more secure than *inner* cipher block chaining, with no material performance penalty.

cipher suite

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

cipher suite name

Cipher suites describe the kind of cryptographics protection that is used by connections in a particular session.

ciphertext

Message text that has been encrypted.

cleartext

Unencrypted plain text.

client

A user, software application (such as a browser), or computer that requests the services, data, or processing of another application or computer (the **server**). A client relies on a service.

cluster

Two or more connected Oracle Application Server instances and the applications deployed to the OC4J instances within those application servers.

confidentiality

A function of cryptography. Confidentiality guarantees that only the intended recipient of a message can view the message (decrypt the ciphertext).

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination **service** and network route information. The destination service is indicated by using its service name for Oracle databases. The network route provides, at a minimum, the location of the **listener** through use of a network address. See **connect identifier**.

connect identifier

A **connect descriptor** or a name that maps to a connect descriptor. A connect identifier can be a **net service name**, database **service name**, or **net service alias**. Users initiate a connect request by passing a username and password along with a connect identifier in a connect string for the service to which they want to connect:

```
CONNECT username/password@connect_identifier
```

connect string

Information the user passes to a **service** to connect to a specific database instance. The information can include username, password, and **net service name**. For example:

```
CONNECT username/password@net_service_name
```

credentials

A username, password, or certificate used to gain access to Oracle Database, Oracle Application Server, or the Oracle Identity Management infrastructure.

CRL

See **certificate revocation lists**.

CRL Distribution Point

An optional extension specified by the X.509 version 3 certificate standard, which indicates the location of the Partitioned CRL where revocation information for a certificate is stored. Typically, the value in this extension is in the form of a URL. CRL distribution points (CRL DP) allow revocation information within a single **certificate authority** domain to be posted in multiple CRLs. CRL DPs subdivide revocation information into more manageable pieces to avoid proliferating voluminous CRLs, thereby providing performance benefits. For example, a CRL DP is specified in the certificate and can point to a file on a Web server from which that certificate's revocation information can be downloaded.

CRL DP

See **CRL Distribution Point**.

cryptography

The practice of encoding and decoding data, resulting in secure messages.

data dictionary

A set of read-only tables that provide information about a database.

Data Encryption Standard (DES)

The U.S. data encryption standard.

database alias

See **net service name**.

decryption

The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).

DES

See **Data Encryption Standard (DES)**.

Diffie-Hellman key negotiation algorithm

A method that lets two parties communicating over an insecure channel to agree upon a random number known only to them. Though the parties exchange information over the insecure channel during execution of the Diffie-Hellman key negotiation algorithm, it is computationally infeasible for an attacker to deduce the random number they agree upon by analyzing their network communications. Oracle Advanced Security uses the Diffie-Hellman key negotiation algorithm to generate session keys.

digital signature

A digital signature is created when a public key algorithm is used to sign the sender's message with the sender's private key. The digital signature assures that the document is authentic, has not been forged by another entity, has not been altered, and cannot be repudiated by the sender.

directory naming

A naming method that resolves a database service, **net service name**, or **net service alias** to a **connect descriptor** stored in a central directory server.

directory naming context

A subtree which is of significance within a directory server. It is usually the top of some organizational subtree. Some directories only permit one such context which is fixed; others permit none to many to be configured by the directory administrator.

distinguished name (DN)

The unique name of an **LDAP**-based directory entry. A distinguished name comprises all of the individual names of the parent entries back to the root.

domain

Any tree or subtree within the **Domain Name System (DNS)** namespace; a group of computers whose host names share a common suffix, the domain name.

Domain Name System (DNS)

A system for naming computers and network services that is organized into a hierarchy of **domain**s. DNS is used in TCP/IP networks to locate computers through user-friendly names. DNS resolves a friendly name into an IP address, which is understood by computers.

encrypted text

Text that has been encrypted, using an encryption algorithm; the output stream of an encryption process. On its face, it is not readable or decipherable, without first being subject to **decryption**. Also called **ciphertext**. Encrypted text ultimately originates as **plaintext**.

encryption

The process of disguising a message rendering it unreadable to any but the intended recipient.

entry

In the context of a directory service, an entry is the building block of a directory. An entry is a collection of information about an object in the directory. Each entry is composed of a set of attributes that describe one particular trait of the object. For

example, if a directory entry describes a person, that entry can have attributes such as first name, last name, telephone number, or e-mail address.

external authentication

Verification of a user identity by a third party authentication service, such as **Kerberos**.

failover

The ability to reconfigure a computing system to utilize an alternate active component when a similar component fails.

Federal Information Processing Standard (FIPS)

A U.S. government standard that defines security requirements for cryptographic modules—employed within a security system protecting unclassified information within computer and telecommunication systems. Published by the **National Institute of Standards and Technology (NIST)**.

FIPS

See **Federal Information Processing Standard (FIPS)**.

grid computing

A computing architecture that coordinates large numbers of servers and storage to act as a single large computer. Oracle Grid Computing creates a flexible, on-demand computing resource for all enterprise computing needs. Applications running on the Oracle 10*g* grid computing infrastructure can take advantage of common infrastructure services for failover, software provisioning, and management. Oracle Grid Computing analyzes demand for resources and adjusts supply accordingly.

group

A collection of OC4J instances that belong to the same cluster topology. Configuration operations can be executed simultaneously on all running OC4J instances in the group.

HTTP

Hypertext Transfer Protocol. The underlying format used by the Web to format and transmit messages and determine what actions Web servers and browsers should take in response to various commands. HTTP is the protocol used between Oracle Application Server and clients.

HTTP server

A **server** that receives HTTP requests from remote browsers, converts the requested URL to a filename, and returns the file to the requester.

HTTPS

Secure Hypertext Transfer Protocol. A protocol that uses the **Secure Sockets Layer (SSL)** as a sublayer under the regular**HTTP** application layer to encrypt and decrypt user page requests as well as the pages that are returned by the origin server.

identity

The combination of the public key and any other public information for an entity. The public information may include user identification data such as an e-mail address. A user certified as being the entity it claims to be.

identity management

The creation, management, and use of online, or digital, entities. Identity management involves securely managing the full life cycle of a digital identity from creation (provisioning of digital identities) to maintenance (enforcing organizational policies regarding access to electronic resources), and, finally, to termination.

identity management realm

A subtree in Oracle Internet Directory, including not only an **Oracle Context**, but also additional subtrees for users and groups, each of which are protected with access control lists.

IIOP

Internet inter-ORB protocol. An Internet transport protocol used by CORBA objects to communicate with each other. In the context of Oracle Application Server, IIOP is used by Java and EJB-based applications. IIOP is also used between Oracle Application Server components.

infrastructure services

A comprehensive deployment platform designed to streamline application deployment by leveraging a single security, directory, and product metadata framework for all applications. The framework includes components such as Oracle Identity Management and OracleAS Metadata Repository.

instance

The set of processes required to run the configured components within an application server installation. There can be only one application server instance for each application server installation. The terms installation and instance are sometimes used interchangeably; however, it is important to remember that an installation is the set of files installed into an Oracle home and an instance is a set of processes associated with those files.

integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

Java Database Connectivity (JDBC)

An industry-standard Java interface for connecting to a relational database from a Java program, defined by Sun Microsystems.

JDBC

See **Java Database Connectivity (JDBC)**.

Kerberos

A network authentication service developed under Massachusetts Institute of Technology's Project Athena that strengthens security in distributed environments. Kerberos is a trusted third-party authentication system that relies on shared secrets and assumes that the third party is secure. It provides **single sign-on (SSO)** capabilities and database link authentication (MIT Kerberos only) for users, provides centralized password storage, and enhances PC security.

key

When encrypting data, a key is a value which determines the **ciphertext** that a given algorithm will produce from given plaintext. When decrypting data, a key is a value

required to correctly decrypt a ciphertext. A ciphertext is decrypted correctly only if the correct key is supplied.

With a symmetric encryption algorithm, the same key is used for both encryption and decryption of the same data. With an asymmetric encryption algorithm (also called a public-key encryption algorithm or public-key cryptosystem), different keys are used for encryption and decryption of the same data.

key pair

A **public key** and its associated **private key**. See **public and private key pair**.

LDAP

See **Lightweight Directory Access Protocol (LDAP)**.

ldap.ora file

A file that contains the following directory server access information:

- Type of directory server
- Location of the directory server
- Default identity management realm or Oracle Context (including ports) that the client or server will use

Lightweight Directory Access Protocol (LDAP)

A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard directory products, such as the Oracle Internet Directory.

listener

A process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server. A listener can be an HTTP server that handles incoming requests and routes them to the dispatcher.

Every time a client requests a network session with a server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the server.

man-in-the-middle

A security attack characterized by the third-party, surreptitious interception of a message, wherein the third party, the *man-in-the-middle*, decrypts the message, re-encrypts it (with or without alteration of the original message), and re-transmits it to the originally-intended recipient—all without the knowledge of the legitimate sender and receiver. This type of security attack works only in the absence of **authentication**.

message digest

Representation of text as a string of single digits. It is created using a formula called a one-way hash function, which is an algorithm that turns a message into a single string of digits. One-way means that it is almost impossible to derive the original message from the string of digits. The calculated message digest can be compared with the message digest that is decrypted with a **public key** to verify that the message has not been tampered with.

middle tier

In a three-tier architecture, the middle tier is the application logic layer. The middle tier provides the computing power and resources for the client. In Oracle Application Server, the middle-tier consists of components such as Oracle HTTP Server, OC4J, and OPMN.

National Institute of Standards and Technology (NIST)

An agency within the U.S. Department of Commerce responsible for the development of security standards related to the design, acquisition, and implementation of cryptographic-based security systems within computer and telecommunication systems, operated by a Federal agency or by a contractor of a Federal agency or other organization that processes information on behalf of the Federal Government to accomplish a Federal function.

net service alias

An alternative name for a **directory naming** object in a directory server. A directory server stores net service aliases for any defined **net service name** or database service. A net service alias entry does not have connect descriptor information. Instead, it only references the location of the object for which it is an alias. When a client requests a directory lookup of a net service alias, the directory determines that the entry is a net service alias and completes the lookup as if it was actually the entry it is referencing.

net service name

The name used by clients to identify a database server. A net service name is mapped to a port number and protocol. Also known as a **connect string**, or **database alias**.

network authentication service

A means for authenticating clients to servers, servers to servers, and users to both clients and servers in distributed environments. A network authentication service is a repository for storing information about users and the services on different servers to which they have access, as well as information about clients and servers on the network. An authentication server can be a physically separate machine, or it can be a facility co-located on another server within the system. To ensure availability, some authentication services may be replicated to avoid a single point of failure.

network listener

A listener on a server that listens for connection requests for one or more databases on one or more protocols. See **listener**.

NIST

See **National Institute of Standards and Technology (NIST)**.

non-repudiation

Incontestable proof of the origin, delivery, submission, or transmission of a message.

obfuscation

A process by which information is scrambled into a non-readable form, such that it is extremely difficult to de-scramble if the algorithm used for scrambling is not known.

object class

A named group of **attribute**s. When you want to assign attributes to an entry, you do so by assigning to that entry the object classes that hold those attributes. All objects associated with the same object class share the same attributes.

Oracle Context

An entry in an LDAP-compliant internet directory called `cn=OracleContext`, under which all Oracle software relevant information is kept, including entries for checksumming security.

There can be one or more Oracle Contexts in a directory. An Oracle Context is usually located in an **identity management realm**.

Oracle PKI certificate usages

Defines the purpose of the key contained in an **certificate**. Oracle PKI certificate usages are based on the key usages defined in the X.509 Version 3 standard.

PCMCIA cards

Small credit card-sized computing devices that comply with the Personal Computer Memory Card International Association (PCMCIA) standard. These devices, also called PC cards, are used for adding memory, modems, or as hardware security modules. PCMCIA cards that are used as hardware security modules securely store the private key component of a **public and private key pair** and some also perform the cryptographic operations as well.

peer identity

SSL connect sessions are between a particular client and a particular server. The identity of the peer may have been established as part of session setup. Peers are identified by **X.509 certificate chain**s.

PEM

The Internet Privacy-Enhanced Mail protocols standard, adopted by the Internet Architecture Board to provide secure electronic mail over the Internet. The PEM protocols provide for encryption, authentication, message integrity, and key management. PEM is an inclusive standard, intended to be compatible with a wide range of key-management approaches, including both symmetric and public-key methods to encrypt data-encrypting keys. The specifications for PEM come from four Internet Engineering Task Force (IETF) documents: RFCs 1421, 1422, 1423, and 1424.

PKCS #10

An RSA Security, Inc., Public-Key Cryptography Standards (PKCS) specification that describes a syntax for certification requests. A certification request, also referred to as a **certificate request**, consists of a distinguished name, a public key, and optionally a set of attributes, collectively signed by the entity requesting certification.

PKCS #11

An RSA Security, Inc., Public-Key Cryptography Standards (PKCS) specification that defines an application programming interface (API), called Cryptoki, to hardware devices which hold cryptographic information and perform cryptographic operations. See also **PCMCIA cards**.

PKCS #12

An RSA Security, Inc., Public-Key Cryptography Standards (PKCS) specification that describes a transfer syntax for storing and transferring personal authentication credentials—typically in a format called a **wallet**.

PKI

See **public key infrastructure (PKI)**.

plaintext

Message text that has not been encrypted.

private key

In public-key cryptography, this key is the secret key. It is primarily used for decryption but is also used for encryption with digital signatures. See **public and private key pair**.

proxy authentication

A process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its *proxy*. The middle tier logs into the directory as a *proxy user*. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

public and private key pair

A set of two numbers used for **encryption** and **decryption**, where one is called the **private key** and the other is called the **public key**. Public keys are typically made widely available, while private keys are held by their respective owners. Though mathematically related, it is generally viewed as computationally infeasible to derive the private key from the public key. Public and private keys are used only with asymmetric encryption algorithms, also called public-key encryption algorithms, or public-key cryptosystems. Data encrypted with either a public key or a private key from a **key pair** can be decrypted with its associated key from the key pair. However, data encrypted with a public key cannot be decrypted with the same public key, and data enwrapped with a private key cannot be decrypted with the same private key.

public key

In public-key cryptography, this key is made public to all. It is primarily used for encryption but can be used for verifying signatures. See **public and private key pair**.

public key encryption

The process where the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using its private key.

public key infrastructure (PKI)

Information security technology utilizing the principles of public key cryptography. Public key cryptography involves encrypting and decrypting information using a shared public and private key pair. It provides for secure, private communications within a public network.

realm

1. Short for **identity management realm**. 2. A **Kerberos** object. A set of clients and servers operating under a single key distribution center/ticket-granting service (KDC/TGS). Services in different realms that share the same name are unique.

realm Oracle Context

An **Oracle Context** that is part of an **identity management realm** in Oracle Internet Directory.

registry

A Windows repository that stores configuration information for a computer.

remote computer

A computer on a network other than the local computer.

remote OC4J instance

OC4J instances other than the **administration OC4J instance**.

restriction

A security scheme that restricts access to files provided by the server to client machines within certain groups of IP addresses or DNS domains.

root key certificate

See **trusted certificate**.

schema

1. Database schema: A named collection of objects, such as tables, views, clusters, procedures, packages, attributes, object classes, and their corresponding matching rules, which are associated with a particular user. 2. **LDAP** directory schema: The collection of attributes, object classes, and their corresponding matching rules.

Secure Sockets Layer (SSL)

An industry standard protocol designed by Netscape Communications Corporation for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI).

server

1. Oracle Application Server, which is a collection of middleware services and tools that provide a scalable, robust, secure, and extensible platform for distributed, object-oriented applications. Oracle Application Server provides a single, unified platform for Java and J2EE, Web Services, XML, SQL, and PL/SQL. 2. Oracle Database Server, which is a relational database server dedicated to performing data management duties on behalf of clients using any number of possible interfaces.

service

1. A network resource used by clients; for example, Oracle Application Server or Oracle database server. 2. An executable process installed in the Windows **registry** and administered by Windows. Once a service is created and started, it can run even when no user is logged on to the computer.

service name

A logical representation of a database, which is the way a database is presented to clients. A database can be presented as multiple services and a service can be implemented as multiple database instances. The service name is a string that is the global database name, that is, a name comprising the database name and domain name, entered during installation or database creation.

session key

A key shared by at least two parties (usually a client and a server) that is used for data encryption for the duration of a single communication session. Session keys are typically used to encrypt network traffic; a client and a server can negotiate a session key at the beginning of a session, and that key is used to encrypt all network traffic

between the parties for that session. If the client and server communicate again in a new session, they negotiate a new session key.

single key-pair wallet

A **PKCS #12**-format **wallet** that contains a single user **certificate** and its associated **private key**. The **public key** is imbedded in the certificate.

single sign-on (SSO)

The ability of a user to *authenticate once,* combined with strong authentication occurring transparently in subsequent connections to other databases or applications. Single sign-on lets a user access multiple accounts and applications with a single password, entered during a single connection. *Single password, single authentication.*

smart card

A plastic card (like a credit card) with an embedded integrated circuit for storing information, including such information as user names and passwords, and also for performing computations associated with authentication exchanges. A smart card is read by a hardware device at any client or server.

A smartcard can generate random numbers which can be used as one-time use passwords. In this case, smartcards are synchronized with a service on the server so that the server expects the same password generated by the smart card.

sniffer

A device used to surreptitiously listen to or capture private data traffic from a network.

SSL

See **Secure Sockets Layer (SSL)**.

SSO

See **single sign-on (SSO)**.

system identifier (SID)

A unique name for an Oracle instance. To switch between Oracle databases, users must specify the desired SID. The SID is included in the CONNECT DATA part of the **connect descriptor** in a tnsnames.ora file, and in the definition of the **network listener** in a listener.ora file.

token card

A device for providing improved ease-of-use for users through several different mechanisms. Some token cards offer one-time passwords that are synchronized with an authentication service. The server can verify the password provided by the token card at any given time by contacting the authentication service. Other token cards operate on a challenge-response basis. In this case, the server offers a challenge (a number) which the user types into the token card. The token card then provides another number (cryptographically-derived from the challenge), which the user then offers to the server.

trusted certificate

A trusted certificate, sometimes called a root key certificate, is a third party identity that is qualified with a level of trust. The trusted certificate is used when an identity is being validated as the entity it claims to be. Typically, the certificate authorities you trust are called trusted certificates. If there are several levels of trusted certificates, a

trusted certificate at a lower level in the certificate chain does not need to have all its higher level certificates reverified.

trusted certificate authority

See **certificate authority**.

trust point

See **trusted certificate**.

user search base

The node in the LDAP directory under which the user resides.

wallet

A wallet is a data structure used to store and manage security credentials for an individual entity. A **Wallet Resource Locator** (WRL) provides all the necessary information to locate the wallet.

wallet obfuscation

An **obfuscation** used to store and access an Oracle **wallet** without querying the user for a password prior to access (supports **single sign-on (SSO)**).

Wallet Resource Locator

(WRL) A locator that provides all necessary information to locate a **wallet**. It is a path to an operating system directory that contains a wallet.

Windows native authentication

An **authentication method** that enables a client single login access to a Windows server and a database running on that server.

WRL

See **Wallet Resource Locator**.

X.509

An industry-standard specification for digital **certificate**s.

Index