Oracle® Containers for J2EE

Configuration and Administration Guide

10*g* Release 3 (10.1.3)

Part No. B14432-01

January 2006

ORACLE®

Oracle Containers for J2EE Configuration and Administration Guide, 10*g* Release 3 (10.1.3)

Part No. B14432-01

Primary Author: Dan Hynes

Contributing Authors: Brian Wright, Sheryl Maring, Bonnie Vaughn

Contributors: Bryan Atsatt, Ellen Barnes, Julie Basu, Steve Button, Olivier Caudron, Jose Alberto Fernandez, Lars Ewe, Marcelo Goncalves, Sumathi Gopalakrishnan, Ping Guo, Hal Hildebrand, James Kirsch, Sunil Kunisetty, Clement Lai, Mike Lehmann, Phlippe Le Mouel, Angie Long, Sharon Malek, Sheryl Maring, Kuassi Mensah, Jasen Minton, Rama Notowidigdo, John O'Duinn, Debu Panda, Shiva Prasad, Chaya Ramanujam, Vinay Shukla, Sanjay Singh, Gael Stevens, Kenneth Tang, Helen Zhao, Frances Zhao, Serge Zloto

Contents

Preface xi

Intended Audience xi
Documentation Accessibility xi
Related Documents xii
Conventions xiii

1 Introducing OC4J

What Is OC4J? 1-1
J2EE Support in OC4J 1-2
What's New or Changed in OC4J 1-2
New Features in OC4J 1-2
Support for Web Services 1-3
Support for New J2EE 1.4 Application Management and Deployment Specifications 1-3
Support for Enterprise JavaBeans 3.0 1-3
Support for Oracle Application Server TopLink 1-3
OracleAS Job Scheduler 1-4
New Two-Phase Commit Transaction Coordinator Functionality 1-4
Generic JMS Resource Adapter Enhancements 1-4
Changes from Previous Releases 1-4
Configuration File Changes 1-4
What Is OC4J in a Standalone Configuration? 1-6
What Is OC4J in an Oracle Application Server Configuration? 1-6
Understanding the Application Hierarchy in OC4J 1-8
The `System` Application 1-8
The `Default` Application 1-9
The Global Web Application 1-9
J2EE Applications 1-9

2 Installing Standalone OC4J

Standalone OC4J Installation Prerequisites 2-1
Installing the Standalone OC4J Distribution 2-2

3 Tools for Administering OC4J

Oracle Enterprise Manager 10*g* Application Server Control Console 3-1
Accessing Application Server Control Console in Standalone OC4J 3-1

Accessing Application Server Control Console in Oracle Application Server 3-2
Functional Overview of the Application Server Control Console Interface 3-2
The admin_client.jar Command Line Utility 3-3
The admin.jar Command Line Utility 3-4
The oc4j/oc4j.cmd Executable Scripts 3-4
Oracle Process Manager and Notification Server (OPMN) 3-4
Changing the oc4jadmin Account Password 3-6

4 OC4J Runtime Configuration

Specifying the JDK Version 4-1
Specifying the JDK in a Standalone Configuration 4-1
Specifying the JDK in a Managed Configuration 4-1
Setting OC4J Runtime Options at Startup 4-2
Setting Runtime Options in a Standalone OC4J Configuration 4-2
Setting Runtime Options in a Managed OC4J Configuration 4-2
Overview of OC4J Runtime Options 4-3
Setting System Properties at Startup 4-4
Setting System Properties in a Standalone OC4J Configuration 4-4
Setting System Properties in an OPMN-Managed OC4J Configuration 4-4
Overview of General System Properties 4-5
Overview of Debug Properties 4-7
Managing stdout/stderr Log Files 4-8

5 Starting and Stopping OC4J

Starting OC4J in a Standalone Environment 5-1
Starting OC4J with oc4j.jar 5-1
Starting OC4J with oc4j/oc4j.cmd 5-2
Starting OC4J in an Oracle Application Server Environment 5-2
Stopping OC4J in a Standalone Environment 5-2
Stopping OC4J with admin.jar 5-2
Stopping OC4J with oc4j/oc4j.cmd 5-3
Stopping OC4J in an Oracle Application Server Environment 5-3
Restarting an OC4J Instance in a Standalone Environment 5-3
Restarting an OC4J Instance in an Oracle Application Server Environment 5-3

6 Using the admin_client.jar Utility

Overview of admin_client.jar Usage 6-1
Understanding the admin_client.jar Syntax and URI Specification 6-2
Performing Operations on a Group of OC4J Instances Within a Cluster 6-2
Performing Operations on a Specific OC4J Instance 6-3
Performing Operations on a Standalone OC4J Server 6-3
Validating a URI 6-4
Printing Usage Text to the Console 6-4
Enabling Logging 6-4
Deploying an Archive 6-5
Deploying a J2EE Application (EAR) 6-5

Deploying a Standalone Web Module (WAR) ... 6-7
Deploying a Standalone Resource Adapter (RAR) ... 6-8
Binding Web Modules to a Web Site Post-Deployment ... 6-9
Bind All Web Modules to a Single Web Site ... 6-9
Bind a Specific Web Module to a Specific Web Site and Set the Context Root ... 6-9
Redeploying an Archive ... 6-10
Undeploying an Archive ... 6-11
Undeploying an EAR or Standalone WAR ... 6-11
Undeploying a Standalone RAR ... 6-11
Updating Modified Classes Only in a Deployed EJB Module ... 6-11
Creating and Managing Shared Libraries ... 6-12
Installing a Shared Library ... 6-12
Modifying an Existing Shared Library ... 6-13
Viewing the Contents of a Shared Library ... 6-14
List All Shared Libraries ... 6-14
Starting/Stopping/Restarting an Application ... 6-14

7 Using the admin.jar Utility

Overview of admin.jar Usage ... 7-1
Understanding the admin.jar Syntax ... 7-2
Printing Help to the Console ... 7-2
Managing a Standalone OC4J Instance ... 7-2
Stopping and Restarting OC4J in a Standalone Environment ... 7-2
Forcing OC4J to Check for Modified Files ... 7-3
Deploying/Undeploying Applications ... 7-3
Managing Applications ... 7-6
Starting/Stopping/Restarting an Application ... 7-6
Updating an EJB Module Within an Application ... 7-7
Managing Web Sites ... 7-8
Managing Data Sources ... 7-10
Creating an Application-Specific Data Source ... 7-10
Listing/Testing/Removing Existing Data Sources ... 7-11
Converting Existing Data Sources to the New Configuration ... 7-12
Deploying/Undeploying Connectors ... 7-12

8 Configuring and Managing Clusters

Clustering Overview ... 8-1
How Clustering Works ... 8-2
Supported Clustering Models ... 8-3
Changes in Clustering ... 8-4
Configuring a Cluster ... 8-5
Configuring Dynamic Node Discovery Using Multicast ... 8-5
Configuring Multicast Discovery with opmnassociate ... 8-8
Configuring Multicast Discovery with opmnctl ... 8-9
Configuring Static Discovery Servers ... 8-11
Configuring a Static Discovery Server Connection with opmnctl ... 8-13

Configuring Cross-Topology Gateways 8-14
Configuring Static Node-to-Node Communication 8-17
Viewing the Status of a Cluster 8-19
Viewing Cluster Status with opmnctl 8-19
Viewing Cluster Status in Application Server Control Console 8-20
Load Balancing with Oracle HTTP Server 8-21
Using Web Server Routing IDs to Control OC4J Request Routing 8-22
Configuring Application Mount Points 8-25
Enabling Dynamic Configuration of Application Mount Points 8-26
Changing the Mount Point Configuration Algorithm 8-27
Viewing Mount Point Configuration Data 8-29
Replicating Changes Across a Cluster 8-30
Creating and Managing Additional OC4J Instances 8-35
Creating an Additional OC4J Instance 8-35
Accessing and Managing a New Instance 8-38
Removing an OC4J Instance 8-39

9 Application Clustering in OC4J

What Is Clustering in OC4J? 9-1
How Does Clustering Differ from Previous OC4J Releases? 9-1
"Islands" No Longer Supported 9-1
loadbalancer.jar No Longer Used 9-2
Deprecated Clustering-Specific XML Elements 9-2
Configuring Application Clustering 9-2
Enabling Clustering 9-2
Setting Replication Policies 9-3
Managing the Number of Nodes to Replicate To 9-5
Synchronous versus Asynchnronous Replication 9-5
Configuring Multicast Replication 9-5
Using an Existing JavaGroups Configuration for Multicast Replication 9-6
Configuring Peer-to-Peer Replication 9-6
Configuring Dynamic OPMN-Managed Peer-to-Peer Replication 9-7
Configuring Static Peer-to-Peer Replication 9-7
Configuring Database Replication 9-9
Disabling Clustering 9-10
Overview of the <cluster> Element 9-10

10 Logging in OC4J

Log Files Generated by OC4J 10-1
Using Plain Text File Logging 10-3
Enabling/Disabling Text File Logging 10-3
Managing Text Log Files 10-4
Viewing Text Log Files 10-4
Using Oracle Diagnostic Logging (ODL) 10-4
Enabling/Disabling ODL 10-4
Managing ODL Log Files 10-6
Viewing ODL Log Files 10-6

Configuring OC4J Logging 10-6
Using and Configuring the OC4J Component Loggers 10-6
Viewing the OC4J Log File 10-7
Configuring the oracle Logger 10-8

11 Task Manager and Thread Pool Configuration

Configuring the OC4J Task Manager 11-1
Using Thread Pools 11-1
Using the Default Thread Pool Configuration 11-2
Managing Thread Pool Configurations 11-2

12 Using MBeans in OC4J

MBeans and Java Management Extensions (JMX) Support in OC4J 12-1
What Are MBeans? 12-1
Overview of the Top-Level OC4J System MBeans 12-2
When Do Changes Made Via MBeans Take Effect? 12-4
How Is MBean Data Persisted? 12-4
Using the System MBean Browser 12-5
Using JMX Notifications 12-5
Subscribing to Notifications 12-5
Using Application-Specific MBeans 12-6

13 Managing Web Sites in OC4J

What Is a Web Site in OC4J? 13-1
Configuring Web Site Connection Data 13-2
Configuring Web Site Data in a Standalone OC4J Installation 13-2
Configuring Web Site Data in OPMN-Managed OC4J Instances 13-2
Configuring Web Sites with opmnctl 13-4
Creating a New Web Site in OC4J 13-5
Creating the Web Site Configuration File 13-6
Referencing the Web Site Configuration File in server.xml 13-7
Defining the Web Site Connection Data in opmn.xml 13-7
Sharing Web Applications Between Web Sites 13-8
Specifying the Cookie Domain 13-9
Configuring a Secure Web Site in OC4J 13-9
Creating the Secure Web Site Configuration File 13-10
Requiring Client Authentication 13-11
Requesting Client Authentication with OC4J 13-11
Starting/Stopping Web Sites 13-12
Configuring Web Site Access Logging 13-12
Configuring Text-Based Access Logging 13-13
Viewing Text Access Log Files 13-14
Configuring ODL Access Logging 13-14
Viewing ODL Access Log Files 13-15
Enabling/Disabling Access Logging for a Web Module/Application 13-15

14 Registering DTDs and XSDs with OC4J

Why Do DTDs/XSDs Have to Be Registered? 14-1
Registering a DTD or XSD 14-2

A Troubleshooting OC4J

Problems and Solutions A-1
java.lang.OutOfMemory Errors A-1
Application Performance Impacted by Garbage Collection Pauses A-2
Invalid or Unneeded Library Elements Degrading Performance A-2
ClassCastExceptions and ClassNotFound Errors A-3
OC4J Fails to Start: Unable to Find Java Compiler A-3
Error Occurs When Clustering an Application A-3
Error When Downgrading from JDK 5.0 to JDK 1.4.2 ..... A-4
OC4J Hangs When Starting Applications in Oracle Application Server A-4
Need More Help? A-5

B Configuration Files Used in OC4J

Overview of the XML Configuration Files Used by OC4J B-1
Overview of the OC4J Server Configuration File (server.xml) B-4
Example of a server.xml File B-5
<application-server> B-6
<application> B-7
<code-source> B-7
<execution-order> B-8
<global-application> B-8
<global-thread-pool> B-8
<global-web-app-config> B-9
<import-shared-library> B-10
<init-param> B-10
<j2ee-logging-config> B-11
<java-compiler> B-11
<javacache-config> B-12
<jms-config> B-12
<log> B-12
<rmi-config> B-13
<shared-library> B-13
<shutdown-class> B-13
<startup-class> B-14
<transaction-manager-config> B-14
<web-site> B-14
Overview of the Web Site Configuration Files (*-web-site.xml) B-15
<web-site> B-15
<description> B-17
<frontend> B-17
<web-app> B-17
<default-web-app> B-19

<user-web-apps> B-20
<access-log> B-20
<odl-access-log> B-20
<ssl-config> B-20

C Overview of the Session State Tables

D Third Party Licenses

ANTLR D-1
The ANTLR License D-1
Apache D-1
The Apache Software License D-2
Apache SOAP D-6
Apache SOAP License D-7
DBI Module D-10
Perl Artistic License D-10
Preamble D-10
Definitions D-10
FastCGI D-12
FastCGI Developer's Kit License D-12
Module mod_fastcgi License D-13
Info-ZIP Unzip Package D-13
The Info-ZIP Unzip Package License D-14
JSR 110 D-14
Jaxen D-14
The Jaxen License D-14
JGroups D-15
The GNU License D-15
mod_mm and mod_ssl D-22
OpenSSL D-23
OpenSSL License D-23
Perl D-25
Perl Kit Readme D-25
mod_perl 1.29 License D-26
mod_perl 1.99_16 License D-27
Perl Artistic License D-30
Preamble D-30
Definitions D-30
SAXPath D-32
The SAXPath License D-32
W3C DOM D-33
The W3C License D-33

Index

Preface

This documentation serves as the primary reference on configuring and managing Oracle Containers for J2EE (OC4J) in both standalone and OPMN-managed (Oracle Application Server) environments. It essentially replaces the *Oracle Application Server Containers for J2EE User's Guide* and the *Oracle Application Server Containers for J2EE Standalone User's Guide* released with previous versions of OC4J.

This preface contains the following sections:

- Intended Audience
- Documentation Accessibility
- Related Documents
- Conventions

Intended Audience

This documentation is intended for the following audiences:

- A systems administrator responsible for configuring and managing an OC4J installation
- A Java application developer using OC4J in a standalone environment

The documentation assumes that readers are already familiar with the following:

- The Java 2 Platform, Enterprise Edition (J2EE) environment
- General server and system administration concepts
- General Web technology
- The Java programming language

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, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the following Oracle resources.

Additional OC4J documents:

- *Oracle Containers for J2EE Deployment Guide*

  This covers information and procedures for deploying an application to an OC4J environment. This includes discussion of the deployment plan editor that comes with Oracle Enterprise Manager 10*g*.

- *Oracle Containers for J2EE Developer's Guide*

  This discusses items of general interest to developers writing an application to run on OC4J—issues that are not specific to a particular container such as the servlet, EJB, or JSP container. (An example is class loading.)

- *Oracle Containers for J2EE Servlet Developer's Guide*

  This provides information for servlet developers regarding use of servlets and the servlet container in OC4J, including basic servlet development and use of JDBC and EJBs.

- *Oracle Containers for J2EE Support for JavaServer Pages Developer's Guide*

  This provides information about JavaServer Pages development and the JSP implementation and container in OC4J. This includes discussion of Oracle features such as the command-line translator and OC4J-specific configuration parameters.

- *Oracle Containers for J2EE JSP Tag Libraries and Utilities Reference*

  This provides conceptual information as well as detailed syntax and usage information for tag libraries, JavaBeans, and other Java utilities provided with OC4J.

- *Oracle Containers for J2EE Services Guide*

  This provides information about standards-based Java services supplied with OC4J, such as JTA, JNDI, JMS, JAAS, and the Oracle Application Server Java Object Cache.

- *Oracle Containers for J2EE Security Guide*

This document (not to be confused with the *Oracle Application Server 10g Security Guide*), describes security features and implementations particular to OC4J. This includes information about using JAAS, the Java Authentication and Authorization Service, as well as other Java security technologies.

- *Oracle Containers for J2EE Enterprise JavaBeans Developer's Guide*

  This provides information about Enterprise JavaBeans development and the EJB implementation and container in OC4J.

- *Oracle Containers for J2EE Web Services Developer's Guide*

  This describes Web services development and configuration in OC4J and Oracle Application Server.

- *Oracle Application Server Advanced Web Services Developer's Guide*

  This book describes topics beyond basic Web service assembly. For example, it describes how to diagnose common interoperability problems, how to enable Web service management features (such as reliability, auditing, and logging), and how to use custom serialization of Java value types.

  This book also describes how to employ the Web Service Invocation Framework (WSIF), the Web Service Provider API, message attachments, and management features (reliability, logging, and auditing). It also describes alternative Web service strategies, such as using JMS as a transport mechanism.

- *Oracle Application Server Web Services Security Guide*

  This describes Web services security and configuration in OC4J and Oracle Application Server.

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.

1

Introducing OC4J

This chapter provides a general introduction to Oracle Containers for J2EE, or OC4J. It includes the following sections:

- What Is OC4J?
- What's New or Changed in OC4J
- What Is OC4J in a Standalone Configuration?
- What Is OC4J in an Oracle Application Server Configuration?
- Understanding the Application Hierarchy in OC4J

What Is OC4J?

Oracle Containers for J2EE 10*g* Release 3 (10.1.3), or OC4J, provides a complete Java 2 Enterprise Edition (J2EE) 1.4-compliant environment. OC4J provides all the containers, APIs, and services mandated by the J2EE specification.

OC4J is distributed in two configurations:

- A **standalone** configuration, in which OC4J is installed as a single, "standalone" instance and is managed, started and stopped directly as a self-contained component.

  See "What Is OC4J in a Standalone Configuration?" on page 1-6 for details on this configuration.

- A **managed** configuration, in which OC4J is installed and managed as a component of Oracle Application Server.

  At a minimum, a managed OC4J installation will include Oracle Process Manager and Notification Server (OPMN), which manages the various Oracle Application Server components, including OC4J.

  An installation will typically also include at least one Oracle HTTP Server (OHS) instance, which provides Web communication and load balancing functionality.

  See "What Is OC4J in an Oracle Application Server Configuration?" on page 1-6 for details.

OC4J is written entirely in Java and executes on the Java Virtual Machine (JVM) of the standard Java Development Kit (JDK). The current OC4J release can run on JDK releases 1.4.2 and 5.0. For standalone OC4J, the JDK must be provided; for OPMN-managed OC4J, the JDK 5.0 is packaged with the server binaries.

The OC4J documentation assumes that you have a basic understanding of Java programming, J2EE technology, and Web and EJB application technology. This includes deployment conventions such as the `/WEB-INF` and `/META-INF` directories.

J2EE Support in OC4J

OC4J supports and is certified on the following standard J2EE specifications, as listed in Table 1–1.

Table 1–1 Supported J2EE Specifications

J2EE Specification	Version Supported By OC4J
JavaServer Pages (JSP)	2.0
Servlets	2.4
Enterprise JavaBeans (EJB)	2.1
	3.0 (Requires JDK 5.0. Support is based on the Enterprise JavaBeans 3.0 Early Draft Review specification.)
Java Management Extensions (JMX)	1.2
J2EE Management	1.0
J2EE Application Deployment	1.1
Java Transaction API (JTA)	1.0
Java Message Service (JMS)	1.1
Java Naming and Directory Interface (JNDI)	1.2
Java Mail	1.2
Java Database Connectivity (JDBC)	3.0
Java Authentication and Authorization Service (JAAS) Provider	1.0
J2EE Connector Architecture	1.5
Enterprise Web Services	1.1
Java API for XML-Based RPC (JAX-RPC)	1.1
SOAP with Attachments API for Java (SAAJ)	1.2
Java API for XML Processing (JAXP)	1.2
Java API for XML Registries (JAXR)	1.0.5

What's New or Changed in OC4J

The following topics outline new features in Oracle Containers for J2EE Release 3 (10.1.3) as well as functional changes from previous releases.

New Features in OC4J

Oracle Containers for J2EE Release 3 (10.1.3) includes a number of new features and enhancements, as described in the following topics:

- Support for Web Services
- Support for New J2EE 1.4 Application Management and Deployment Specifications

- Support for Enterprise JavaBeans 3.0
- Support for Oracle Application Server TopLink
- OracleAS Job Scheduler
- New Two-Phase Commit Transaction Coordinator Functionality
- Generic JMS Resource Adapter Enhancements

Support for Web Services

OC4J provides full support for Web services in accordance with the J2EE 1.4 standard, including JAX-RPC 1.1. Web services interoperability is also supported.

- Support for the Enterprise Web Services 1.1 specification
- EJB 2.1 Web services end point model
- JSR 109 client and server deployment model
- CORBA Web services: Support for wrapping existing basic CORBA Servants as Web services and auto-generating WSDL from IDL
- Support for source code annotations to customize Web services behavior such as invocation and ending styles (RPC/literal, RPC/encoded, Doc/literal); customizing the Java to XML mapping; enforcing security.
- Database and JMS Web services

Support for New J2EE 1.4 Application Management and Deployment Specifications

OC4J supports the following specifications defining new standards for deploying and managing applications in a J2EE environment.

- The *Java Management Extensions (JMX) 1.2* specification, which allows standard interfaces to be created for managing resources, such as services and applications, in a J2EE environment. The OC4J implementation of JMX provides a JMX client that can be used to completely manage an OC4J server and applications running within it.
- The *J2EE Management Specification (JSR-77)*, a specification that allows standard components to be created for managing applications in a J2EE environment.
- The *J2EE Application Deployment API (JSR-88)*, which defines a standard API for configuring and deploying J2EE applications and modules into a J2EE-compatible environment. The OC4J implementation includes the ability to create and/or edit a deployment plan containing the OC4J-specific configuration data needed to deploy a component into OC4J.

Support for Enterprise JavaBeans 3.0

OC4J provides support for Enterprise JavaBeans 3.0, including the new program annotation functionality, as defined in the Early Draft Review specification. The specification is available at the following link:

```
http://java.sun.com/products/ejb/
```

Note that OC4J must use JDK 5.0 to enable EJB 3.0 support.

Support for Oracle Application Server TopLink

Oracle Application Server TopLink is an advanced, object persistence framework for use with a wide range of Java 2 Enterprise Edition (J2EE) and Java application

architectures. OracleAS TopLink includes support for the OC4J Container Managed Persistence (CMP) container and base classes that simplify Bean Managed Persistence (BMP) development.

OracleAS Job Scheduler

The OracleAS Job Scheduler provides asynchronous scheduling services for J2EE applications. Its key features include capabilities for submitting, controlling and monitoring *jobs*, defined as a unit of work that executes when the work is performed.

New Two-Phase Commit Transaction Coordinator Functionality

The new Distributed Transaction Manager in OC4J can coordinate two-phase transactions between any type of XA resource, including databases from Oracle as well as other vendors and JMS providers such as IBM WebSphere MQ. Automatic transaction recovery in the event of a failure is also supported.

Generic JMS Resource Adapter Enhancements

The Generic JMS Resource Adapter can now be used as an OC4J plug-in for OracleAS JMS that ships with the current version of OC4J as well as for IBM WebSphere MQ JMS version 5.3.

Support for lazy transaction enlistment has been added so that JMS connections can be cached and still be able to correctly participate in global transactions.

Finally, the Generic JMS Resource Adapter now has better error handling. Endpoints now automatically retry after provider or system failures, and `onMessage()` errors are handled correctly.

Changes from Previous Releases

The following changes have been made in Oracle Containers for J2EE 10*g* Release 3 (10.1.3). Functional changes in Oracle Application Server 10*g* Release 3 (10.1.3) that pertain to OC4J are also outlined.

Configuration File Changes

The following changes have been made to configuration files utilized in standalone OC4J and in OC4J instances installed as components of Oracle Application Server. All of the files noted are installed in *`ORACLE_HOME`*`/j2ee/`*`instance`*`/config` by default.

application.xml

- The `<persistence>` element has been moved to the new `system-application.xml` file.
- The `<jazn>` element now points to the new `system-jazn-data.xml` file as the security configuration file for the OC4J instance.
- The `default-data-source` attribute of the root `<orion-application>` element now specifies "jdbc/OracleDS" as the default data source in both standalone OC4J and Oracle Application Server.
- The `<ejb-module>` element for "PortComponentLinkResolver" has been removed.
- The `<odl>` element, used to enable ODL logging for the `default` application, has been added but commented out as a subelement of `<log>`.

ascontrol-web-site.xml

- This file has been removed from both standalone OC4J and Oracle Application Server. The Application Server Control Console instance deployed to OC4J is now bound to `default-web-site.xml` by default and is accessible via the `/em` context root.

default-web-site.xml

- This file configures the default Web site used in both standalone OC4J and Oracle Application Server. All applications, including the Application Server Control Console deployed to the OC4J instance, are accessed by default through the `default` Web site using the context root specified in this file.

global-web-application.xml

- The `<dtd>` element has been removed from the Oracle Application Server version of this file.
- The `<url-pattern>` element in the rmi-tunnel servlet definition specifies `rmiTunnel/*` in both standalone OC4J and Oracle Application Server.

http-web-site.xml

- This file has been removed from both standalone OC4J and Oracle Application Server. All applications deployed to the OC4J instance are now bound to `default-web-site.xml` by default.

j2ee-logging.xml

- This new file is used to configure Java Loggers, including the `oracle` Logger.

jazn-data.xml

- This file no longer contains the security configuration for the OC4J instance. This configuration is now defined in the new `system-jazn-data.xml` file. It can be specified at the application level to define users and roles, however.

oc4j-connectors.xml

- The `location` attribute of the `<connector>` element is no longer specified for the "datasources" and "OracleASjms" connectors.

server.xml

- The `<web-site>` elements pointing to `http-web-site.xml` and `ascontrol-web-site.xml` have been removed. A single element now points to `default-web-site.xml`, the configuration file for the `default` Web site.
- Multiple `<shared-library>` elements have been added, each referencing a shared library installed with OC4J.

system-application.xml

- This is a new file, added to provide configuration for the `system` application. See "The `System` Application" on page 1-8 for more information on this new internal component.

system-jazn-data.xml

- This new file contains the security configuration for the OC4J instance. It essentially replaces `jazn-data.xml`.

What Is OC4J in a Standalone Configuration?

The standalone or "unmanaged" OC4J configuration offers a robust J2EE-compliant container that is easy to administer. In this configuration, a single OC4J instance is installed into a single `ORACLE_HOME` - the root directory in which Oracle software is installed.

The standalone OC4J configuration is comprised of the following components:

- Oracle Containers for J2EE 10*g* Release 3 (10.1.3)
- Oracle Enterprise Manager 10*g* Application Server Control Console, a Web-based administration application installed by default with OC4J

The Application Server Control Console is enabled immediately upon installation. See "Oracle Enterprise Manager 10g Application Server Control Console" on page 3-1 for details on using this management interface.

Installation

The standalone OC4J distribution, which includes Application Server Control Console, is provided as a ZIP archive. See Chapter 2, "Installing Standalone OC4J" for instructions.

Administration

The OC4J instance is administered as a standalone component, using either the Application Server Control Console installed with the instance or the built-in `admin.jar` or `oc4j.cmd/oc4j.sh` command line utilities.

See Chapter 3, "Tools for Administering OC4J" for an overview of these tools.

Starting/Stopping

In a standalone configuration, an OC4J instance is started using either the `oc4j.jar` command line or the `oc4j.cmd/oc4j.sh` command line utilities. Startup options and system properties can also be set at startup on the `oc4j.jar` command line.

See "Starting OC4J in a Standalone Environment" on page 5-1 for details.

Backup, Restore and Disaster Recovery Capabilities

The OC4J standalone configuration does not have backup, restore and disaster recovery capabilities.

Web Communication

Web communication in a standalone environment is provided through the built-in OC4J Web server, which supports HTTP and HTTPS communications natively without the use of the Oracle HTTP Server (OHS).

The default Web site is defined in the `default-web-site.xml` file, which specifies the default HTTP listener on port `8888`. Additional Web sites may be defined on different ports using variations of this file. See Chapter 13, "Managing Web Sites in OC4J" for instructions on creating additional Web sites in OC4J.

What Is OC4J in an Oracle Application Server Configuration?

In this configuration, OC4J is installed as a component of Oracle Application Server. A typical configuration includes the following components:

- Oracle Containers for J2EE 10*g* Release 3 (10.1.3)

- Oracle Enterprise Manager 10*g* Application Server Control Console, a Web-based administration application installed by default with OC4J
- Oracle HTTP Server (OHS) 1.3, which provides front-end Web communication and load balancing functionality
- Oracle Process Manager and Notification Server (OPMN), used to start/stop and monitor the other installed components, including OC4J and OHS. OMPN includes Oracle Notification Server (ONS), which manages communications between components.

Oracle Application Server provides support for HTTP session and stateful session Enterprise JavaBean replication and load balancing across a cluster of OC4J instances. See Chapter 9, "Application Clustering in OC4J" for details.

The Oracle Universal Installer provides a number of installation options:

- **Integrated Web Server, J2EE Server and Process Management**

  In this configuration, all components - OC4J, OHS and OPMN - are installed into a single `ORACLE_HOME` directory.

  Multiple OC4J instances can be created within this `ORACLE_HOME` directory. Multiple host machines, each hosting one or more OC4J instances, can be included in an Oracle Application Server cluster.

- **J2EE Server and Process Management**

  This installation includes OC4J and OPMN. It can be utilized as a "standalone" OPMN-managed OC4J instance for development or testing purposes, or can be included within an Oracle Application Server cluster.

- **Web Server and Process Management**

  This installation includes OHS and OPMN only. It can be used as a "standalone" OHS instance, typically serving as the front-end Web listener for an Oracle Application Server cluster.

Installation

Installation of the various components is done using the Oracle Universal Installer. Note that OPMN must be installed in every `ORACLE_HOME` directory to enable monitoring of each installed component.

Administration

All administration tasks are performed using the Web-based Application Server Control Console user interface. In a clustered environment, a single Application Server Control Console can be used to manage all OC4J instances in the cluster.

See "Oracle Enterprise Manager 10g Application Server Control Console" on page 3-1 for details on this application.

Starting/Stopping

In a managed environment, OPMN is used to start and stop all components, including OC4J. See "Starting OC4J in an Oracle Application Server Environment" on page 5-2 for details.

Note that OC4J runtime options and system properties can be manually set in the OPMN configuration file, `opmn.xml`. See Chapter 4, "OC4J Runtime Configuration" for details.

Backup, Restore and Disaster Recovery Capabilities

These capabilities are available with the managed Oracle Application Server configuration.

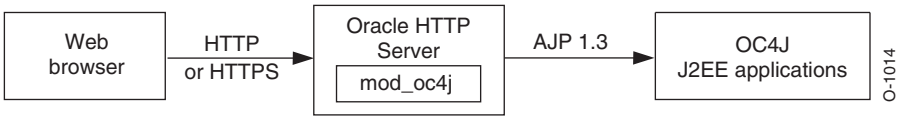
Web Communication

A standalone OPMN-managed OC4J instance (the J2EE Server and Process Manager install type) can use the built-in OC4J Web server to directly receive and respond to HTTP[S] requests.

Web communications with OC4J can also be managed through Oracle HTTP Server (OHS), which serves as a front-end listener, and the `mod_oc4j` module, which forwards HTTP requests to OC4J server instances using the Apache JServ Protocol (AJP) 1.3.

The OHS-OC4J request/response flow is as follows:

1. An incoming HTTP request is received by the OHS listener.
2. OHS passes the request to an OC4J server instance via the `mod_oc4j` module. The connection between OHS and OC4J uses the Apache JServ Protocol (AJP) on a port number negotiated during OC4J startup.



Mount points mapping request URLs to the OC4J instance serving the requested application are dynamically created in `mod_oc4j` at the time applications are deployed. Requests that come in for specific mount points are routed to the OC4J instance corresponding to that mount point.

For additional information on configuring and managing OHS and the `mod_oc4j` module, see the *Oracle HTTP Server Administrator's Guide*.

Understanding the Application Hierarchy in OC4J

This section provides an overview of the application hierarchy within an OC4J instance.

The `System` Application

The `system` application is a new internal component in Oracle Containers for J2EE Release 3 (10.1.3). It is auto-deployed to the OC4J instance the first time OC4J is started.

This application was added primarily to address issues related to deploying or redeploying applications to OC4J. It sits at the root of the application hierarchy, and provides classes and configuration required at OC4J startup. For example, it provides the shared libraries imported by default by all other deployed applications, such as the Oracle JDBC driver and XML parser implementations.

The `system` application is an OC4J internal component only. Applications cannot be deployed to it, nor can it be declared the parent of another application. The `default` application continues to serve as the default parent of all deployed applications.

The configuration for the system application is defined in system-application.xml, which is installed in *ORACLE_HOME*/j2ee/*instance*/config by default.

Important: Because system is a key internal component that is critical to OC4J startup, the system-application.xml file should NOT be modified.

The only exception is the <jazn> tag, which may be modified as needed to specify changes to the security provider and/or location of the OC4J security configuration file (system-jazn-data.xml).

The Default Application

The default application sits just below system in the application hierarchy. It continues to serve as the default parent of all other J2EE applications deployed into the OC4J instance. As such, all configuration parameters defined for the default application are inherited by all other applications, unless explicitly overridden at the application level.

Standalone Web modules (WAR files) may also be deployed to the default application.

The configuration for the default application is defined in application.xml, which is installed in *ORACLE_HOME*/j2ee/*instance*/config by default.

The Global Web Application

The global Web application is the Web module component of the default application. It provides configuration data applied by default to all Web modules deployed to the OC4J instance. It also contains initialization parameters applied by default to all servlets.

The configuration file for the default Web application is global-web-application.xml, which is installed in *ORACLE_HOME*/j2ee/*instance*/config by default. This file contains parameters that are applied by default to all Web modules deployed to the OC4J instance, as well as servlet initialization parameters applied to all servlets. Any of these parameter values can be overridden by corresponding values in a Web module's orion-web.xml file, however.

In a standalone OC4J installation, the root directory of the default Web application is j2ee/home/default-web-app. To deploy to the default Web application, place your JSP pages and class files under this directory in the standard Web application directory structure: static pages and JSP pages at the top level, servlet classes under j2ee/home/default-web-app/WEB-INF/classes, and library JAR files in j2ee/home/default-web-app/WEB-INF/lib.

J2EE Applications

By default, an application deployed to an OC4J instance inherits configuration parameters from its designated parent application, or from the default application if no other parent is specified. However, a parameter value set in an application's orion-application.xml descriptor overrides an equivalent parameter inherited from the parent.

Note that a Web module must be contained within a parent J2EE application. A WAR file is typically packaged and deployed with the EAR file that defines the parent J2EE application. However, a WAR file can be deployed to the `default` application as a standalone Web module.

2

Installing Standalone OC4J

This chapter describes the prerequisites and process for installing the OC4J standalone distribution, which is distributed as the as `oc4j_extended.zip` archive.

For instructions on installing OC4J as a component of Oracle Application Server, see the platform-specific *Oracle Application Server Installation Guide*.

The following topics are covered in this chapter:

- Standalone OC4J Installation Prerequisites
- Installing the Standalone OC4J Distribution

Standalone OC4J Installation Prerequisites

Ensure the following prerequisites are met before installing a standalone OC4J server.

Install JDK 1.4.2 or 5.0

Before installing standalone OC4J, you must first install Java 2 Platform, Standard Edition (J2SE) Development Kit (JDK) release 1.4.2 or 5.0 on the OC4J host machine. You can download the JDK release from `http://java.sun.com/j2se/`.

> **Note:** For standalone OC4J, the JDK must be provided; for OPMN-managed OC4J, the JDK 5.0 is packaged with the server binaries.

Set Environment Variables

After installing J2SE, ensure that the following environment variables are set:

Table 2–1 Environment Variable Settings

Environment Variable	Value
`JAVA_HOME`	Set to the location of the JDK. This variable is required to start the OC4J server. Note that the JDK that will be used must be added to the host machine's PATH.
`ORACLE_HOME`	Set to the root directory into which you will install the OC4J distribution. Defining this variable is required if you intend to run an `oc4j.cmd` or `oc4j.sh` executable script. For example, if you install OC4J into `C:\oracle`, set the value of the `ORACLE_HOME` variable to this directory.

Table 2–1 (Cont.) Environment Variable Settings

Environment Variable	Value
J2EE_HOME	Optionally create and set this variable to *oc4j_install_dir*/j2ee/home, the installed location of oc4j.jar and admin.jar. Setting this variable will allow you to invoke these JAR files from any directory.

Installing the Standalone OC4J Distribution

Install the standalone OC4J distribution by extracting the oc4j_extended.zip file into the directory that will serve as the OC4J installed directory - referenced in this documentation as *ORACLE_HOME* - using the archive utility of your choice. The installer automatically creates the required directory structure for you, as shown below.

```
ORACLE_HOME
   /bin
   /j2ee
   /javacache
   /javavm
   /jdbc
   /jdk
   /jlib
   /lib
   /rdbms
   /soap
   /sqlj
   /toplink
   /webservices
```

You will be prompted to set a password for the OC4J Administrator account the first time OC4J is started. The user name for this account is set to oc4jadmin by default.

Note that you can also activate the oc4jadmin account before starting OC4J using the jazn.jar tool. This tool is also located in the ORACLE_HOME/j2ee/home/ directory. The syntax is as follows:

```
jazn.jar -activateadmin password
```

Note: The oc4j.jar -install command, previously used to activate the oc4jadmin account as well as set the password for this account, is deprecated in Oracle Containers for J2EE 10*g* Release 3 (10.1.3).

The OC4J standalone distribution is installed with a default configuration that includes a default Web site where applications can be accessed, and a Web site that allows the Application Server Control management interface to be used. These are provided so that you can start using OC4J immediately. See Chapter 13, "Managing Web Sites in OC4J" for additional information.

3

Tools for Administering OC4J

This chapter provides an overview of the administrative capabilities provided with OC4J. It includes the following sections:

- Oracle Enterprise Manager 10g Application Server Control Console
- The admin_client.jar Command Line Utility
- The admin.jar Command Line Utility
- The oc4j/oc4j.cmd Executable Scripts
- Oracle Process Manager and Notification Server (OPMN)

Oracle Enterprise Manager 10*g* Application Server Control Console

The *Oracle Enterprise Manager 10g Application Server Control Console* is a JMX-compliant, Web-based user interface for deploying, configuring and monitoring applications within OC4J, as well as managing the OC4J server instance and the Web services used by your applications. This section covers the following topics:

- Accessing Application Server Control Console in Standalone OC4J
- Accessing Application Server Control Console in Oracle Application Server
- Functional Overview of the Application Server Control Console Interface

See the online Help provided with Application Server Control Console for detailed instructions on using this interface.

> **Note:** The current release of Application Server Control Console does not provide management support for either OPMN or Oracle HTTP Server. Use the OPMN command-line tool, `opmnctl`, to start, stop and manage instances of these components.

Accessing Application Server Control Console in Standalone OC4J

The Application Server Control Console is installed and configured automatically when you install the OC4J software. It is started by default when OC4J is started.

The console is accessed through the `default` Web site, which is configured to listen for HTTP requests on port `8888`. To access the console, simply type the following URL in a Web browser:

`http://`*`hostname`*`:8888/em`

Accessing Application Server Control Console in Oracle Application Server

The Application Server Control Console is installed and configured automatically when you install OC4J using the Oracle Universal Installer.

The console is started with all other installed Oracle Application Server components using the OPMN command-line tool, `opmnctl`, which is installed in the *`ORACLE_HOME`*`/opmn/bin` directory on each server node. Start all installed components by issuing the following command:

```
cd ORACLE_HOME/opmn/bin
opmnctl startall
```

In a typical Oracle Application Server installation, all Web applications, including Application Server Control Console, are accessed through Oracle HTTP Server (OHS). Use the following URL to access the console:

```
http://ohs_host_address:port/em
```

- *`ohs_host_address`* is the address of the OHS host machine.; for example, `server07.company.com`
- *`port`* is an HTTP listener port assigned to OHS by OPMN. Run the following `opmnctl` command on the OHS host machine to get the list of assigned listener ports from OPMN:

  ```
  opmnctl status -l
  ```

  Supply the port designated as `http1` in the OPMN status output as the value for *`port`*:

  ```
  HTTP_Server | HTTP_Server | 6412 | Alive | 1970872013 | 1
  6396 | 0:48:01 | https1:4443,http2:722,http1:7779
  ```

Functional Overview of the Application Server Control Console Interface

The Application Server Control Console is organized into several functional areas, described below.

Applications

- Start/stop applications, modules or standalone resource adapters deployed into the OC4J instance
- Deploy, undeploy or redeploy an application or module
- Create or edit a deployment plan as part of deploying an application
- View statistics on HTTP requests and active EJB method calls

Administration

- Manage J2EE services, including JMS and JTA
- View and search for JNDI names
- Create JDBC data sources and connection pools providing database access
- Set JSP container properties
- Configure security providers and manage users and roles
- Access MBeans through the JMX MBean browser
- Subscribe to event-driven JMX notifications

Performance

- View graphs showing usage of CPU and memory resources by OC4J versus other active applications, as well as OC4J heap usage
- View statistics on database connections and transaction activity, JVM usage, JSP and servlet requests and EJB methods
- Query system for most-requested JSPs, servlets and EJBs

Web Services

- Enable or disable a Web service
- View metrics and statistics for Web services running within the instance
- View the WSDL for a Web service
- Test a Web service
- Configure auditing, logging, reliability and security for a Web service

Logs

- View log files for specific applications deployed into the OC4J instance
- View logs for the default (global) application and Application Server Control Console
- Search logs for specific message types and strings
- View XML formatted log files for components using the Oracle Diagnostic Logging (ODL) framework
- Retrieve Web service logs

See Chapter 10, "Logging in OC4J" for more on the logging capabilities provided by OC4J.

The admin_client.jar Command Line Utility

OC4J provides a command-line utility— `admin_client.jar`—that can be used to perform operations on active OC4J instances in an Oracle Application Server clustered environment as well as on standalone OC4J servers.

Among the tasks you can perform with this utility:

- Deploy an application (EAR), a standalone Web module (WAR), a standalone EJB module (EJB JAR) or a standalone resource adapter (RAR) to a specific OC4J instance or to all instances within a cluster
- Undeploy an application, Web module, EJB module or resource adapter
- 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 Chapter 6, "Using the admin_client.jar Utility" for instructions on using this tool.

The admin.jar Command Line Utility

OC4J provides a command-line utility called `admin.jar` that can be used to perform operations on an active standalone OC4J instance. Among other things, you can use this utility to:

- Shut down and restart a standalone OC4J instance
- Restart a specific application
- Deploy or undeploy applications to a standalone OC4J instance
- Add or remove a Web site
- Add, remove or test a global or application-specific data source

The utility is installed by default in `ORACLE_HOME/j2ee/home`. Note that OC4J must be started before this utility can be used. Also note that the utility cannot be used to start OC4J. See Chapter 7, "Using the admin.jar Utility" for instructions on using this tool.

The oc4j/oc4j.cmd Executable Scripts

The OC4J distribution includes executable scripts - a shell script for the Unix/Linux platforms and a batch file for the Windows platform - that can be used in an OC4J standalone configuration to start and stop a local OC4J instance, get the OC4J version, and complete the OC4J installation process.

The `oc4j` executable scripts are located in the `ORACLE_HOME/bin` directory. The scripts are platform-specific:

- Use the `oc4j` shell script on Unix and Linux platforms.
- Use the `oc4j.cmd` batch file on Windows platforms.

Both executables use the same syntax, which is as follows:

```
oc4j [options]
```

The set of options that can be passed to the executables is identical for both, as summarized below.

Table 3–1 Options for oc4j executables

Option	Description
`-start`	Starts the OC4J instance.
`-shutdown` `-port` *`ormiport`* `-password` *`password`*	Stops the OC4J instance. `-port` *`ormiport`*: You do not need to specify the port if OC4J is running on the default ORMI port, which is `23791`. `-password` *`password`*: Specify the `oc4jadmin` account password.
`-version`	Returns the OC4J version number.

Oracle Process Manager and Notification Server (OPMN)

In a managed OC4J environment, OPMN is used to manage as well as start and stop all installed Oracle Application Server components, including all OC4J instances. OPMN also monitors OC4J and associated components, such as OHS. As a result,

OPMN must be installed into each ORACLE_HOME to monitor installed Oracle Application Server components.

See the *Oracle Process Manager and Notification Server Administrator's Guide* for instructions on configuring and using OPMN.

A command-line utility, `opmnctl`, is used to control the OPMN daemon. The utility is installed by default in the `ORACLE_HOME/opmn/bin` directory on any machine hosting Oracle Application Server host components.

> **Note:** The current release of Application Server Control Console does not provide management support for either OPMN or Oracle HTTP Server. Use the OPMN command-line tool, opmnctl, to start/stop and manage instances of these components.

OPMN is configured through the `opmn.xml` configuration file, which is located in the `ORACLE_HOME/opmn/conf` directory. Most edits to this file must be made by hand, as the current release of Application Server Control Console does not provide a file editing capability.

The following is an abridged example of how OC4J configuration data is structured in the `opmn.xml` configuration file.

- Configuration data for each component is set in an `<ias-component>` element, where the `id` attribute equals the component name.; in this case, `OC4J`.
- Each individual OC4J instance created on the host machine is configured within a `<process-type>` element. The `id` attribute uniquely identifies the instance.
- The `<process-set>` element defines a group of OC4J processes created at startup.

  The value of the `id` attribute identifies the group and is appended to log files generated for processes within the group to aid in management.

The following is an abridged example of the OC4J configuration data structure in `opmn.xml`:

```
<opmn>
 ...
 <ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value=" -Djava.awt.headless=true"/>
        <data id="java-bin" value="/jdk/bin"/>
        <data id="oc4j-options" value="-validateXML -verbosity 10"/>
      </category>
      <category id="stop-parameters">
        <data id="java-options" value="-Djava.awt.headless=true"/>
      </category>
    </module-data>
    <start timeout="600" retry="2"/>
    <stop timeout="120"/>
    <restart timeout="720" retry="2"/>
    <port id="default-web-site" protocol="ajp" range="12501-12600"/>
    <port id="rmi" range="12401-12500"/>
    <port id="jms" range="12601-12700"/>
    <process-set id="default_group" numprocs="1"/>
  </process-type>
```

```
 </ias-component>
</opmn>
```

Changing the oc4jadmin Account Password

The OC4J administrator account is created by default with the user name `oc4jadmin`. This account is required to invoke commands using the various tools provided with OC4J, such as the `admin_client.jar` command line utility, and can also be used to log in to Application Server Control Console.

The `oc4jadmin` account is assigned the `oc4j-administrators` role, which an account must have to manage users and roles. An account must also have this role to connect to the JMX MBean server.

The initial password for this account can be set when OC4J is installed; otherwise, you will be prompted to set it the first time OC4J is started. The password can later be changed through the Setup page in Application Server Control Console.

To change the password for the `oc4jadmin` account:

1. Log in to Application Server Control Console as oc4jadmin.
2. If multiple instances are configured, select the instance you want to change the password for. Otherwise, you will change the password for the OC4J instance the Application Server Control Console is running on.
3. Click the **Setup** link located at the bottom of any page in the user interface.
4. Change the password.

4

OC4J Runtime Configuration

This chapter provides details on runtime options and system properties that can be set at OC4J startup. It includes the following topics:

- Specifying the JDK Version
- Setting OC4J Runtime Options at Startup
- Setting System Properties at Startup

Specifying the JDK Version

OC4J requires the Java 2 Platform, Standard Edition (J2SE) Development Kit (JDK) release 1.4.2 or 5.0. You can specify the JDK version to use for a standalone OC4J configuration, as well as for each OC4J instance in an Oracle Application Server installation.

Specifying the JDK in a Standalone Configuration

In a standalone OC4J configuration, set the `JAVA_HOME` environment variable to the location of the JDK you wish to be used by OC4J.

Note that the JDK that will be used must be added to the host machine's PATH.

Alternatively, you can specify the JDK to use at OC4J startup. For example:

```
C:\ORACLE_HOME\j2ee\home\C:\jdk\bin\java -jar oc4j.jar
```

Specifying the JDK in a Managed Configuration

An OPMN-managed OC4J instance installed as a component of Oracle Application Server will use the JDK 5.0 by default. This new version of the JDK is required to utilize EJB 3.0 and offers numerous performance improvements. However, if applications that will be deployed to OC4J require a JDK 1.4.2 release, it may be necessary to "downgrade" to the earlier version.

Before switching from JDK 5.0 to JDK 1.4.2, you must remove all compiled application files from the OC4J instance. To do this:

- Stop the OC4J instance.
- Delete the *ORACLE_HOME*/j2ee/*instance*/application-deployments directory.

  Deleting this directory will cause the application files to be re-compiled when OC4J is re-started with the JDK 1.4.2.

You can specify the JDK to use for each OC4J instance through manual edits to the `opmn.xml` configuration file.

Set Java system properties in the `<data>` element where the `id` attribute is `"java-bin"`. This `<data>` element is enclosed within the `<category id="start-parameters">` subelement of the `<ias-component id="OC4J">` element in the XML structure. For example:

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-bin" value="c:/myhost/jdk/bin/java"/>
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

Setting OC4J Runtime Options at Startup

A number of OC4J runtime options can be set on OC4J instances at OC4J startup, most providing options for managing standard output messages. How these options are set differs for standalone OC4J and managed Oracle Application Server configurations.

- Setting Runtime Options in a Standalone OC4J Configuration
- Setting Runtime Options in a Managed OC4J Configuration
- Overview of OC4J Runtime Options

Setting Runtime Options in a Standalone OC4J Configuration

OC4J runtime options can be set by passing arguments on the `oc4j.jar` command line at OC4J startup. The syntax for `oc4j.jar` is as follows:

```
java [props] -jar oc4j.jar [args]
```

Note that runtime options (`[args]`) are specified after `oc4j.jar` in the syntax. For example:

```
java -jar oc4j.jar -validateXML -verbosity 10
```

Setting Runtime Options in a Managed OC4J Configuration

When OC4J is installed as a component of Oracle Application Server, OC4J runtime options must be manually added to the `opmn.xml` configuration file. Options will be passed to managed OC4J instances at startup.

Set Java system properties in the `<data>` element where the `id` attribute is `"oc4j-options"`. This `<data>` element is enclosed within the `<category id="start-parameters">` subelement of the `<ias-component id="OC4J">` element in the XML structure. Preface all system properties with a `-D`. For example:

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="oc4j-options" value="-validateXML -verbosity 10"/>
```

```
      ...
    </category>
    ...
  </module-data>
 </process-type>
</ias-component>
```

Overview of OC4J Runtime Options

Table 4–1 describes the OC4J runtime options.

Table 4–1 OC4J Startup Options

Command-Line Argument	Description
`-quiet`	Suppresses standard output to the console.
`-config` *`path`*	Specifies the path to the `server.xml` descriptor file. The default location is the *`ORACLE_HOME`*`/j2ee/home/config` directory.
`-validateXML`	Validates XML configuration files defined by an XSD at the time they are read.

`-out [`*`file`*`]`	Specifies a file to route the standard output to. The file contains messages that are printed to `System.out`, as well as the messages sent to output through the servlet logging interface. If not specified, all output is written to standard out.
	See "Managing stdout/stderr Log Files" on page 4-8 for additional system properties that can be set to manage `stdout` files.
	Note that in an OPMN-managed configuration, the file will be generated within an *instance*`_default_group_1` directory appended to the path specified.
	For example, if you specify the following in `opmn.xml`:
	`<data id="oc4j-options" value=" ... -out /mypath/mylog.log" />`
	The `mylog.log` file will actually be generated in:
	`/mypath/instanceName_default_group_1/mylog.log`

`-err [`*`file`*`]`	Specifies a file to route standard error output to. The file contains messages that are printed to `System.err`. If not specified, all errors are written to standard error. See "Managing stdout/stderr Log Files" on page 4-8 for additional system properties that can be set to manage `stderr` files. Note that in an OPMN-managed configuration, the file will be generated within an *instance*`_default_group_1` directory appended to the path specified. See the `-out` description above for details.
`-verbosity` *`int`*	Define an integer between `1` and `10` to set the verbosity level of the message output. A value of `10` will produce the most verbose output. For example: `java -jar oc4j.jar -verbosity 10`
`-monitorResourceThreads`	Enables backup debugging of thread resources. Enable only if you have problems with threads getting stuck in critical sections of code.
`-userThreads`	Enables context lookup support from user-created threads.
`-listProperties`	Outputs a list of all of the OC4J-specific system properties that can be set on the JVM at OC4J startup, then exits. The following example will redirect the output to a text file in the working directory: `java -jar oc4j.jar -listProperties > props.txt`
`-version`	Returns the installed version of OC4J and exits
`-? |-help`	Prints the help text for these options to the console.

Setting System Properties at Startup

You can set a number of OC4J-specific system properties on the JVM at OC4J startup.

- Setting System Properties in a Standalone OC4J Configuration
- Setting System Properties in an OPMN-Managed OC4J Configuration
- Overview of General System Properties
- Overview of Debug Properties
- Managing stdout/stderr Log Files

Note: You can output a list of all of the OC4J-specific system properties that can be set on the JVM at OC4J startup using the `oc4j.jar -listProperties` option. The following example will redirect the output to a text file in the working directory:

```
java -jar oc4j.jar -listProperties > props.txt
```

Setting System Properties in a Standalone OC4J Configuration

You can set system properties on the JVM through the OC4J command line at startup. If OC4J is running, you must restart the instance for new property settings to take effect.

The syntax is as follows:

```
java [props] -jar oc4j.jar [args]
```

Note that all system properties (`[props]`) are specified before `oc4j.jar` in the syntax. All system properties must be prefaced on the command line with a `-D`. For example:

```
java -DGenerateIIOP=true -Dhttp.session.debug=true -jar oc4j.jar
```

Setting System Properties in an OPMN-Managed OC4J Configuration

When OC4J is installed as a component of Oracle Application Server, OC4J system properties must be manually added to the `opmn.xml` configuration file. Options will be passed to managed OC4J instances at startup.

Set Java system properties in the `<data>` element where the `id` attribute is `"java-options"`. This `<data>` element is enclosed within the `<category id="start-parameters">` subelement of the `<ias-component id="OC4J">` element in the XML structure. Preface all system properties with a `-D`. For example:

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="-DGenerateIIOP=true
         -Dhttp.session.debug=true"/>
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

Overview of General System Properties

The following table describes the general system properties that can be set for OC4J.

Table 4–2 -D General System Properties for OC4J

Property	Description
`java.ext.dirs`	Sets the external directories to be searched for classes when compiling.
`java.io.tmpdir=` *`new_tmpdir`*	Sets the temporary directory for the deployment wizard. The default is `/tmp/var`. The deployment wizard uses 20 MB in swap space of the temp directory for storing information during the deployment process. At completion, the deployment wizard cleans up the temp directory. However, if the wizard is interrupted, it may not have the time or opportunity to clean up the temp directory. In this case, you must clean up any additional deployment files from this directory yourself. If not, the directory may fill up, which will disable any further deployment. If you receive an `Out of Memory` error, check for space available in the temp directory.
`GenerateIIOP=` `<true|false>`	Enables IIOP stub generation. The default is `false`.
`KeepIIOPCode=` `<true|false>`	Set whether the generated IIOP stub/tie code is kept. The default is `false`.
`oracle.arraylist.deepCopy=` `<true|false>`	If `true`, then while cloning an array list, a deep copy is performed. If `false`, a shallow copy is performed for the array list. The default is `true`.
`dedicated.rmicontext=` `<true|false>`	This property replaces the deprecated `dedicated.connection` setting. The default is `false`. When two or more clients in the same process retrieve an `InitialContext`, OC4J returns a cached context. Thus, each client receives the same `InitialContext`, which is assigned to the process. Server lookup, which results in server load balancing, happens only if the client retrieves its own `InitialContext`. If you set `dedicated.rmicontext=true`, then each client receives its own `InitialContext` instead of a shared context. When each client has its own `InitialContext`, then the clients can be load balanced. Note that you can also set this in the JNDI properties.
`oracle.mdb.fastUndeploy=` *`int`*	Sets the interval at which OC4J polls the underlying database to check if an MDB session is shut down, in seconds. This property enables you to shut down OC4J cleanly when you are running MDBs in a Windows environment or when the back-end database is running on a Windows environment. Normally when you use an MDB, it is blocked in a receive state waiting for incoming messages. However, if you shut down OC4J while the MDB is in a wait state in a Windows environment, the OC4J instance cannot be stopped and the applications are not undeployed since the MDB is blocked. Setting this property enables OC4J to poll the database to see if the session is shut down when the MDB is not processing incoming messages and in a wait state. If you do not set this property and you try to shutdown OC4J using CTRL-C, then the OC4J process will hang for at least 2.5 hours. Note that this polling process can be expensive for performance, and should not be set to start too frequently.

Table 4–2 (Cont.) -D General System Properties for OC4J

Property	Description
`oracle.dms.sensors=<none|normal|heavy|all>`	You can set the value for Oracle built-in performance metrics to the following: ■ `none`: Disables metrics ■ `normal`: Medium number of metrics (default) ■ `heavy`: High number of metrics ■ `all`: Every possible metric This parameter should be set on the OC4J server. The previous method for turning on these performance metrics, `oracle.dms.gate=<true|false>`, is replaced by the `oracle.dms.sensors` variable. However, if you still use `oracle.dms.gate`, then setting this variable to false is equivalent to setting `oracle.dms.sensors=none`.
`associateUsingThirdTable=<true|false>`	For container-managed relationships in entity beans, you can designate whether a third database table is used to manage the relationship. Set to `false` if you do not want a third association table. The default is `false`.
`DefineColumnType=<true|false>`	Set this to `true` if you are using a pre-9.2.0 Oracle JDBC driver. For these drivers, setting this variable to true avoids a round-trip when executing a select over the Oracle JDBC driver. This parameter should be set on the OC4J server. The default is `false`. When you change the value of this option and restart OC4J, it is only valid for applications deployed after the change. Any applications deployed before the change are not affected. When true, the `DefineColumnType` extension saves a round trip to the database that would otherwise be necessary to describe the table. When the Oracle JDBC driver performs a query, it first uses a round trip to a database to determine the types that it should use for the columns of the result set. Then, when JDBC receives data from the query, it converts the data, as necessary, as it populates the result set. When you specify column types for a query with the `DefineColumnType` extension set to true, you avoid the first round trip to the Oracle database. The server, which is optimized to do so, performs any necessary type conversions.

Table 4–2 (Cont.) -D General System Properties for OC4J

Property	Description
`oc4j.formauth.redirect=<true|false>`	This property is applicable when form-based authentication is used by a Web application. If set to `true`, OC4J will perform a client side redirect back to the request URL after a user enters valid credentials when accessing a resource. If the user does not have valid credentials, the Web browser will be redirected to the form authentication error page defined for the Web application. If set to `false`, the `/j-security-check` URL will be displayed in the browser after the user enters valid credentials. The default is `false`.
`http.proxyHost=`*`proxy_host`* `http.proxyPort=`*`proxy_port`*	If your HTTP traffic goes through a proxy Web server, specify the proxy host and optionallythe proxy port in the command line. If *proxy_port* is omitted, the default is port 80.
`http.webdir.enable=<true|false>`	This property enables or disables servlet class name invocation for all servlets within the OC4J instance. If set to `true`, any servlet running in the OC4J instance can be invoked by class name by default. If set to `false`, servlets cannot be invoked by class name. The default is `false`. To disable this functionality on a per-Web-application basis, set this property to `true`, then set `<orion-web-app servlet-webdir="" ... />` in the `orion-web.xml` descriptor for each Web application that should not allow servlet class name invocation. Note that the value set for `servlet-webdir` in `orion-web.xml` overrides the default value set for this attribute in `ORACLE_HOME/j2ee/home/config/global-web-application`, which is `servlet-webdir="/servlet"`.

Overview of Debug Properties

Note: The debug properties listed in this section are deprecated in Oracle Containers for J2EE 10*g* Release 3 (10.1.3).

See "Using and Configuring the OC4J Component Loggers" on page 10-6 for details on using the component loggers provided with OC4J.

Use the following properties to better debug applications running within OC4J. Debug messages are printed to the console. All properties take a Boolean value.

Preface all properties with a `-D`.

Table 4–3 OC4J Debug Properties

Debug Property	Description
`http.session.debug=<true|false>`	Provides information about HTTP session events to the console.
`http.request.debug=<true|false>`	Provides information about each HTTP request to the console.
`http.cluster.debug=<true|false>`	Provides information about HTTP clustering events to the console.

Table 4–3 (Cont.) OC4J Debug Properties

Debug Property	Description
`http.error.debug=<true|false>`	Prints all HTTP errors to the console.
`http.method.trace.allow=<true|false>`	Enables the `trace` HTTP method.
`datasource.verbose=<true|false>`	Provides verbose information on creation of data source and connections using data sources and connections released to the pool.
`jdbc.debug=<true|false>`	Provides verbose information when JDBC calls are made.
`ejb.cluster.debug=<true|false>`	Enables EJB clustering debug messages.
`rmi.debug=<true|false>`	Prints RMI debug information to the console.
`rmi.verbose=<true|false>`	Provides verbose information on RMI calls.
`jca.connection.debug=<true|false>`	Provides extra diagnostic information for J2CA connections.
`ws.debug=<true|false>`	Enables Web Services debugging.

Managing stdout/stderr Log Files

The following properties are used to manage standard `stderr` and `stdout` log files.

The type of log file(s) that the properties pertain to are specified with the `-out` and/or `-err` command line options. You can also set a location to output these log files to in these options.

For example, set the following to rotate `stdout/stderr` files when the file size reaches 2.5 MB. Log files will be output to the `D:\logs` directory.

```
java -Dstdstream.filesize=2.5 -jar oc4j.jar -out d:\logs\oc4j.out -err
d:\logs\oc4j.err
```

This example will rotate `stdout` logs at 2:30 p.m. every day and limit the archive to a maximum of 10 files:

```
java -Dstdstream.rotatetime=14:30 -Dstdstream.filenumber=10 -jar oc4j.jar -out
d:\logs\oc4j.out
```

Table 4–4 stdout/stderr Archive Management Properties

Debug Property	Description
`stdstream.filesize=`*`max_file_size`*	The maximum size any file in the archive will be allowed to grow to, in megabytes. Files are rotated when this maximum is reached.
`stdstream.filenumber=`*`max_files`*	The maximum number of files to keep as archives. The oldest file will be automatically deleted when the limit is exceeded.
`stdstream.rotatetime=`*`HH:mm`*	The time at which the log file will be rotated each day.

5

Starting and Stopping OC4J

This chapter provides instructions for installing OC4J, as well as for starting, stopping and restarting an OC4J instance. It includes the following sections:

- Starting OC4J in a Standalone Environment
- Starting OC4J in an Oracle Application Server Environment
- Stopping OC4J in a Standalone Environment
- Stopping OC4J in an Oracle Application Server Environment
- Restarting an OC4J Instance in a Standalone Environment
- Restarting an OC4J Instance in an Oracle Application Server Environment

Starting OC4J in a Standalone Environment

You can start an OC4J server instance in a standalone environment using the default configuration with either the `oc4j.jar` command line or one of the `oc4j` executable scripts.

Starting OC4J with oc4j.jar

To start OC4J by invoking `oc4j.jar`, issue the following command from the `ORACLE_HOME/j2ee/home/` directory:

```
java -jar oc4j.jar [args]
```

Invoking `oc4j.jar` as shown above starts OC4J using the default `server.xml` configuration file, which you can find in the `j2ee/home/config` directory. To start OC4J using a non-default version of the `server.xml` file, issue the following command. Note that you must supply the path to the modified configuration file.

```
java -jar oc4j.jar -config /yourpath/server.xml [args]
```

Note that you can optionally pass in arguments at startup to set runtime options in OC4J. For an overview of valid arguments, see "Setting OC4J Runtime Options at Startup" on page 4-2. You can also view the console help by issuing the following command from the `ORACLE_HOME/j2ee/home` directory:

```
java -jar oc4j.jar -help
```

You can also set system properties on the JVM through the `oc4j.jar` command line at OC4J startup. For details on setting system properties, see "Setting System Properties at Startup" on page 4-4.

Starting OC4J with oc4j/oc4j.cmd

To start OC4J using the `oc4j` scripts, issue the following command from the `ORACLE_HOME/bin` directory:

```
oc4j -start
```

Note that the ORACLE_HOME and JAVA_HOME environment variables must be set to use this command. See "Standalone OC4J Installation Prerequisites" on page 2-1 for details.

Starting OC4J in an Oracle Application Server Environment

In a managed configuration, all Oracle Application Server components - including OC4J and OHS - should be started using `opmnctl`, the OPMN command line tool. This tool is installed in the `ORACLE_HOME/opmn/bin` directory.

Use the following command to start all OPMN managed processes, including OC4J, on a local Oracle Application Server instance:

```
opmnctl startall
```

Use the following command to start a specific managed processes - in this case OC4J - on a local Oracle Application Server instance:

```
opmnctl startproc ias-component=OC4J
```

Stopping OC4J in a Standalone Environment

Stop OC4J by invoking the `-shutdown` command in the `admin.jar` command-line utility or an `oc4j.cmd` or `oc4j` executable script.

Note: Operating system commands such as `Control-C` on Windows or `kill` on Unix/Linux machines should not be used to stop OC4J.

This is especially true when applications utilizing EJBs are actively running within OC4J, as such commands do not allow EJB method calls or timer operations to complete before shutting down the server.

Stopping OC4J with admin.jar

To stop OC4J using `admin.jar`, issue the following command:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword -shutdown
[ordinary|force] [reason]
```

Note the following options that can be specified:

- `[ordinary | force]`
  The type of shutdown. The default is `ordinary`, which allows each thread to terminate normally.

  The `force` option terminates all threads immediately. It is essentially the same as unplugging the host machine. If this option is used, the current state for clustered applications will not be replicated.

- `[reason]`
  You can optionally specify a reason for the shutdown as a string that is written to

the *ORACLE_HOME*/j2ee/home/log/server.log file. Spaces are not allowed in the string.

The following example forces a shutdown of the OC4J server using admin.jar, which terminates all threads immediately. Note the string entered as the reason for the shutdown, which is written to the ORACLE_HOME/j2ee/home/config/server.log file.

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -shutdown force
  need_to_reboot_host_machine
```

Stopping OC4J with oc4j/oc4j.cmd

To stop OC4J using one of the oc4j scripts, issue the following command from the ORACLE_HOME/bin directory. Note that you must supply the ORMI port used by OC4J, which is 23791 by default, as well as the password for the oc4jadmin account.

```
oc4j -shutdown -port oc4jOrmiPort -password adminPassword
```

For example:

```
oc4j.cmd -shutdown -port 23791 -password adminpwd
```

Note that the *ORACLE_HOME* and *JAVA_HOME* environment variables must be set to use this command. See "Standalone OC4J Installation Prerequisites" on page 2-1 for details.

Stopping OC4J in an Oracle Application Server Environment

In a managed configuration, all Oracle Application Server components - including OC4J and OHS - should be stopped using opmnctl, the OPMN command-line tool. This tool is installed in the *ORACLE_HOME*/opmn/bin directory.

Use the following command to stop all OPMN-managed processes, including OC4J, on a local Oracle Application Server instance:

```
opmnctl stopall
```

Use the following command to stop a specific managed component - in this case OC4J - on a local Oracle Application Server instance:

```
opmnctl stopproc ias-component=OC4J
```

Restarting an OC4J Instance in a Standalone Environment

You can restart OC4J using the admin.jar command-line utility. Restart OC4J by executing the following command:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId   adminPassword -restart
[reason]
```

Note that you can optionally enter a string as the value for [*reason*]. The string is written to the *ORACLE_HOME*/j2ee/home/config/server.log file.

Restarting an OC4J Instance in an Oracle Application Server Environment

In a managed configuration, all Oracle Application Server components - including OC4J and OHS - should be restarted using opmnctl, the OPMN command line tool. This tool is installed in the ORACLE_HOME/opmn/bin directory.

Use the following command to restart all OPMN managed processes, including OC4J, on a local Oracle Application Server instance:

```
opmnctl startall
```

Use the following command to restart a specific managed processes - in this case OC4J - on a local Oracle Application Server instance:

```
opmnctl restartproc ias-component=OC4J
```

6

Using the admin_client.jar Utility

OC4J provides a command-line utility— `admin_client.jar`—that can be used to perform operations on active OC4J instances in an Oracle Application Server clustered environment as well as on standalone OC4J servers.

Among the tasks you can perform with this utility:

- Deploy an application (EAR), a standalone Web module (WAR), a standalone EJB module (EJB JAR) or a standalone resource adapter (RAR) to a specific OC4J instance or to a "group" of instances within a cluster
- Undeploy an application, Web module, EJB module or resource adapter
- 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

This chapter includes the following topics:

- Overview of admin_client.jar Usage
- Deploying an Archive
- Binding Web Modules to a Web Site Post-Deployment
- Redeploying an Archive
- Undeploying an Archive
- Updating Modified Classes Only in a Deployed EJB Module
- Creating and Managing Shared Libraries
- Starting/Stopping/Restarting an Application

Overview of admin_client.jar Usage

The `admin_client.jar` utility is installed by default in *`ORACLE_HOME`*`/j2ee/home` in an OC4J instance. Note that OC4J must be started before this utility can be used.

Note that this is the preferred tool for performing operations on OC4J.

This section covers the following topics:

- Understanding the admin_client.jar Syntax and URI Specification
- Printing Usage Text to the Console
- Enabling Logging

Understanding the admin_client.jar Syntax and URI Specification

The `admin_client.jar` utility uses the following syntax. The parameters are described in the text after the syntax.

```
java -jar admin_client.jar uri adminId adminPassword command
```

The key parameter passed on the command line is *uri*, which specifies the target for the commands supplied. The syntax for the URI varies depending on the instance(s) being targeted. See the following for the format of this URI:

- Performing Operations on a Group of OC4J Instances Within a Cluster
- Performing Operations on a Specific OC4J Instance
- Performing Operations on a Standalone OC4J Server
- Validating a URI

The OC4J administration user name and password are also passed to the utility. The user name for the default administrator account is `oc4jadmin`.

As an example, the following command will start the `petstore` application, which is installed in the `home` OC4J instance on `node1`, a member of an Oracle Application Server cluster:

```
java -jar admin_client.jar deployer:oc4j:opmn://node1.company.com/home
oc4jadmin password -application petstore -start
```

Performing Operations on a Group of OC4J Instances Within a Cluster

Use the following URI to specify all OC4J instances within a group as the target. A *group* is defined as a loosely syncronized set of like-named OC4J instances within the same cluster topology. For example, all instances named `home` within a cluster would collectively form a group across which configuration operations can be executed simultaneously.

The URI utilizes the OPMN-based clustering framework. You only need to supply the host name and optionally OPMN request port for any Oracle Application Server node within the cluster; the application is then able to retrieve the host names and OPMN ports for all other nodes within the cluster.

The URI syntax is as follows:

```
deployer:cluster:[rmis]:opmn://host[:opmnPort]/oc4jInstanceName
```

For example:

```
deployer:cluster:opmn://node1.company.com/home
```

Table 6–1 URI Parameters for Targeting a Group

Parameter	Description
`rmis`	Optional. Include if the target utilizes ORMI over SSL, or ORMIS.
`host`	Required. The host name of an Oracle Application Server node within the cluster. Any node can be specified; the list of other nodes in the cluster will be retrieved from this node.
`opmnPort`	Optional. The OPMN request port, as specified in `opmn.xml`. If not specified, the default port `6003` will be used.
`oc4jInstanceName`	Required. The common instance name shared by OC4J instances within the group.

Performing Operations on a Specific OC4J Instance

Use the following URI to target a specific OPMN-managed OC4J instance, including an instance within a cluster. Note that `cluster:` is replaced by `oc4j:` in the prefix.

Specify the host name for the Oracle Application Server node hosting the instance. If you are not sure of the host name or port for the node, you can specify the host name for another node within the cluster, as well as the name of the Oracle Application Server instance. The application will then use the OPMN clustering framework to locate the node hosting the Oracle Application Server instance.

The URI syntax is as follows:

```
deployer:oc4j:[rmis]:opmn://host[:opmnPort]/[iASInstanceName]
/oc4jInstanceName
```

For example:

```
deployer:oc4j:opmn://server.company.com:6004/instance2/home
```

Table 6–2 URI Parameters for Targeting a Specific Instance

Parameter	Description
`rmis`	Optional. Include if the target utilizes ORMI over SSL, or ORMIS.
`host`	Required. The host name of the Oracle Application Server node to target within the cluster to use as the OPMN server.
`opmnPort`	Optional. The OPMN request port, as specified in `opmn.xml`. If not specified, the default port `6003` will be used.
`iASInstanceName`	Optional. The name of the Oracle Application Server instance to target, if it does not reside on the node specified for *`host`*.
`oc4jInstanceName`	Required. The name of the target OC4J instance.

Performing Operations on a Standalone OC4J Server

Use the following URIs to target a standalone OC4J server instance.

If using RMI, the URI syntax is as follows:

```
deployer:oc4j:host:rmiPort
```

If using ORMI over SSL (ORMIS), specify the following:

```
deployer:oc4j:rmis:host:ormisPort
```

For example:

```
deployer:oc4j:myserver:23791
deployer:oc4j:rmis:myserver:23943
```

Table 6–3 URI Parameters for Targeting Standalone OC4J

Parameter	Description
`rmis`	Required if the target utilizes ORMI over SSL, or ORMIS.
`host`	Required. The host name of an Oracle Application Server node within the cluster. Any node can be specified; the list of other nodes in the cluster will be retrieved from this node.
`rmiPort`	Required if RMI used. The RMI port, as specified in the instance-specific `rmi.xml` file.

Table 6–3 (Cont.) URI Parameters for Targeting Standalone OC4J

Parameter	Description
ormisPort	Required if ORMIS used. The SSL port, as specified in the instance-specific `rmi.xml` file.

Validating a URI

You can validate a URI using the `-validateURI` command.

```
java -jar admin_client.jar uri adminId adminPassword -validateURI
```

For example:

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/
home oc4jadmin password -validateURI
```

Printing Usage Text to the Console

To print the inline help text for the `admin_client.jar` commands to the console, simply type `-help` on the command line.

To view detailed help for a specific command, type `-usage` followed by the command identifier. For example:

```
java -jar admin_client.jar -usage [command]
```

Enabling Logging

To help troubleshoot errors that occur when running `admin_client.jar`, you can enable Java logging when running this tool. Log messages will be output to the console.

To enable logging:

1. Create a `logging.properties` file containing a single line:

   ```
   oracle.oc4j.admin.jmx.client.CoreRemoteMBeanServer.level=INFO
   ```

   Note that if you create this file in a location other than `ORACLE_HOME/home/j2ee`, you must include the path to the file in the command below.

2. Set `-Djava.util.logging.config.file=logging.properties` on the `admin_client.jar` command line as follows:

   ```
   java -Djava.util.logging.config.file=logging.properties -jar admin_client.jar
   uri adminId adminPassword command
   ```

You can set the value in the `logging.properties` file to one of the following Java log level values:

Table 6–4 Java Log Levels

Java Log Level	Description
`SEVERE`	Log system errors requiring attention from the system administrator.
`WARNING`	Log actions or a conditions discovered that should be reviewed and may require action before an error occurs.

Table 6–4 (Cont.) Java Log Levels

Java Log Level	Description
INFO	Log normal actions or events. This could be a user operation, such as "login completed" or an automatic operation such as a log file rotation.
CONFIG	Log configuration-related messages or problems.
FINE	Log trace or debug messages used for debugging or performance monitoring. Typically contains detailed event data.
FINER	Log fairly detailed trace or debug messages.
FINEST	Log highly detailed trace or debug messages.

For example:

```
oracle.oc4j.admin.jmx.client.CoreRemoteMBeanServer.level=FINE
```

Deploying an Archive

You can use `admin_client.jar` to deploy an application (EAR), a standalone Web module (WAR) or a standalone resource adapter (RAR) to a specific OC4J instance or to all instances within a cluster.

This chapter covers the following:

- Deploying a J2EE Application (EAR)
- Deploying a Standalone Web Module (WAR)
- Deploying a Standalone Resource Adapter (RAR)

Usage Note:

- Deploying an archive across a cluster requires that all instances have the same `oc4jadmin` account password.

Deploying a J2EE Application (EAR)

Use the `-deploy` command to deploy or redeploy a J2EE application packaged as an EAR file. The EAR-specific syntax is as follows:

```
java -jar admin_client.jar uri adminId adminPassword -deploy -file
path/filename-deploymentName appName [-bindAllWebApps [webSiteName]]
[-targetPath path] [-parent appName] [ -deploymentDirectory path]
-enableIIOP [-iiopClientJar path/filename][-deploymentPlan path/filename]
```

Ideally, you should include the `-bindAllWebApps` subswitch to bind all Web modules within the EAR to the Web site they will be accessed through. If no Web site is specified, modules will be bound to the `default` Web site.

For example, the following command deploys the `utility` application to all OC4J `home` instances within the cluster of which `node1` is a member. All Web modules within the application will be bound to the `default` Web site.

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/home
oc4jadmin password -deploy -file C:/dev/utility.ear -deploymentName utility
-bindAllWebApps
```

Table 6–5 -deploy Command Subswitches for EAR Deployment

Subswitch	Description
`-file`	Required. The path and filename of the EAR file to deploy.
`-deploymentName`	Required. The user-defined application deployment name, used to identify the application within OC4J.
`-bindAllWebApps`	Optional. Binds all Web modules to the specified Web site, or to the `default` Web site if none specified. If not specified, you must use the `-bindAllWebApps` command described on on page 6-9. You can optionally supply a value for *`webSiteName`*, which is the *`name`* portion of the *`name`*`_web-site.xml` file that configures the Web site.
`-targetPath`	Optional. The directory to deploy the EAR to. If not specified, the EAR is deployed to the *`ORACLE_HOME`*`/j2ee/`*`instance`*`/applications/` directory by default. The deployed EAR file is also copied to this directory. Each successive deployment will cause this EAR file to be overwritten.
`-parent`	Optional. The parent application of this application. The default is the global or `default` application.
`-deploymentPlan`	Optional. The path and filename for a deployment plan to apply to the application. The plan would have been saved during a previous deployment as an XML file. The file must exist on the local host.
`-deploymentDirectory`	Optional. The directory containing the OC4J-specific deployment descriptors and generated files, such as compiled JSP classes and EJB wrapper classes. The default directory is *`ORACLE_HOME`*`/j2ee/`*`instance`*`/applications/`.
`-sequential`	Optional. Include to deploy the archive to each OC4J instance within the cluster in sequence. Requests will not be routed to an instance while the EAR is being deployed to it. If not included, the archive is simultaneously deployed to all instances by default. This option is valid in a clustered environment only; it is not valid for standalone OC4J.
`-enableIIOP`	Optional. Include to generate IIOP client stubs on the OC4J server. The application-level stubs generated for all EJB modules are output to an archive named `_iiopClient.jar` in the *`ORACLE_HOME`*`/j2ee/home/application-deployments/`*`appName`* directory. In addition, stubs for each individual EJB module are generated in an archive with the same name in the *`ORACLE_HOME`*`/j2ee/home/application-deployments/`*`appName`*`/`*`ejbModuleName`*`/` directory. Note that the `GenerateIIOP` system property must be enabled at OC4J startup to use this feature. This property is set as `-DGenerateIIOP=true` on the OC4J command line for OC4J standalone or as an `oc4j-options` value in `opmn.xml`.

Table 6–5 (Cont.) -deploy Command Subswitches for EAR Deployment

Subswitch	Description
`-iiopClientJar`	Optional. The path and filename of the JAR to output IIOP client stubs to. The application-level stubs generated for all EJB modules are output to an archive named `_iiopClient.jar` in the *ORACLE_HOME*`/j2ee/home/application-deployments/`*appName* directory. If a path is supplied, the archive is also set on this path. In addition, stubs for each individual EJB module are generated in an archive with the same name in the *ORACLE_HOME*`/j2ee/home/application-deployments/`*appName*`/`*ejbModuleName*`/` directory. Note that the `GenerateIIOP` system property must be enabled at OC4J startup to use this feature. This property is set as `-DGenerateIIOP=true` on the OC4J command line for OC4J standalone or as an `oc4j-options` value in `opmn.xml`.

Deploying a Standalone Web Module (WAR)

Use the `-deploy` command to deploy or redeploy a standalone Web module packaged as a WAR file.

The WAR-specific syntax is as follows:

```
java -jar admin_client.jar uri adminId adminPassword -deploy -file
path/filename -deploymentName appName [-bindAllWebApps [webSiteName]]
[-targetPath path] [-parent appName] [-deploymentDirectory path]
[-contextRoot context]
```

The WAR can be designated a child of another deployed application that does not already contain a Web module component; otherwise, it will be deployed to the `default` application.

Note that a WAR cannot be deployed as the child of an application that already contains a Web module. That is, if the `acme` application already contains an `acme-web.war`, an additional WAR file cannot be deployed into that application. Re-package the WAR in the application's EAR file and redeploy the application instead.

The following command deploys the standalone `acme-web.war` Web module to the `default` application across all OC4J `home` instances within the cluster. Because the `-bindAllWebApps` subswitch is included, but a Web site to bind to is not specified, the module will be bound to the `default` Web site.

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/home
oc4jadmin password -deploy -file C:/dev/acme-web.war
-deploymentName utility -bindAllWebApps -parent default
```

Table 6–6 -deploy Command Subswitches for WAR Deployment

Subswitch	Description
`-file`	Required. The path and filename of the archive to deploy.
`-deploymentName`	Required. The user-defined Web module name, used to identify it within OC4J.

Table 6–6 (Cont.) -deploy Command Subswitches for WAR Deployment

Subswitch	Description
`-bindAllWebApps`	Optional. Binds all Web modules to the specified Web site, or to the `default` Web site if none specified. You can optionally supply a value for *`webSiteName`*, which is the *`name`* portion of the *`name`*`_web-site.xml` file that configures the Web site.
`-targetPath`	Optional. The directory to deploy the archive to. If not specified, the archive is deployed to the *`ORACLE_HOME`*`/j2ee/`*`instance`*`/applications/` directory by default. The generated EAR file containing the standalone WAR file is also copied to this directory. Each successive deployment will cause this archive to be overwritten.
`-parent`	Optional. The parent application the module will be deployed to. The default is the `default` application.
`-deploymentDirectory`	Optional. The directory containing the OC4J-specific deployment descriptors and generated files, such as compiled JSP classes. The default directory is *`ORACLE_HOME`*`/j2ee/`*`instance`*`/application-deployments/`.

`-contextRoot`	Optional. The Web module context, which will be appended to the URL used to access the application through a Web browser. If not specified, the value passed in for `-deploymentName` will be used.
	For example, if you supply the following as the context root:
	`/petstore`
	The module could be accessed with the following URL:
	`http://node1.company.com:7777/petstore`

Deploying a Standalone Resource Adapter (RAR)

Use the `-deploy` command to deploy or redeploy a Java Connector Architecture-compliant resource adapter packaged as a RAR file. By default, resource adapters are deployed to the *`ORACLE_HOME`*`/j2ee/`*`instance`*`/connectors/` directory.

Because a standalone RAR is a resource that may be shared by multiple applications, redeploying or undeploying a standalone RAR requires a restart of the `default` application. This in turn will force a restart of all deployed applications, which are child applications of `default`. See "Starting/Stopping/Restarting an Application" on page 6-14 for instructions on restarting an application.

The RAR-specific syntax is as follows:

```
java -jar admin_client.jar uri adminId> adminPassword -deploy -file
path/filename-deploymentName connectorName [-nativePathLib path]
[-grantAllPermissions]
```

The following command deploys the `acme-rar.rar` module to all OC4J `home` instances within the cluster.

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/home
oc4jadmin password -deploy -file /dev/acme-rar.rar -deploymentName acme-rar
-grantAllPermissions
```

Table 6–7 -deploy Command Subswitches for RAR Deployment

Subswitch	Description
`-file`	Required. The path and filename of the RAR file to deploy.
`-deploymentName`	Required. The user-defined connector name, used to identify the connector within OC4J.
`-nativeLibPath`	Optional. The path to the directory containing native libraries (such as DLLs) within the RAR file.
`-grantAllPermissions`	Include to grant all runtime permissions requested by the resource adapter, if required.

Binding Web Modules to a Web Site Post-Deployment

Every Web module deployed to OC4J must be bound to a Web site through which it will be accessed.

Typically, you will bind Web modules packaged as WAR files within an EAR at the time the EAR is deployed using the `-bindAllWebApps` subswitch on the `-deploy` command. However, if the `-bindAllWebApps` subswitch was not specified when the EAR was deployed, you can bind modules to a Web site post-deployment, as the following topics describe:

- Bind All Web Modules to a Single Web Site
- Bind a Specific Web Module to a Specific Web Site and Set the Context Root

Bind All Web Modules to a Single Web Site

Use the `-bindAllWebApps` command to bind all Web modules within a J2EE application to the same Web site, or to `default-web-site` by default.

The syntax is:

```
java -jar admin_client.jar uri adminId adminPassword -bindAllWebApps
-appName appName> -webSiteName siteName
```

Table 6–8 -bindAllWebApps Command Subswitches

Subswitch	Description
`-appName`	Required. The name of the parent application as specified at deployment time.
`-webSiteName`	Optional. The name of the *name*`_web-site.xml` file that denotes the Web site to bind the Web modules to. If omitted, all modules are bound to `default-web-site`.

Bind a Specific Web Module to a Specific Web Site and Set the Context Root

Use the `-bindWebApp` command to bind a single Web module within a J2EE application to a specific Web site, or to `default-web-site` by default. You can also optionally set the context root that will be used to access the module.

The syntax is:

```
java -jar admin_client.jar uri adminId adminPassword -bindWebApp
-appName appName -webModuleName moduleName -webSiteName siteName
-contextRoot contextRoot
```

Table 6–9 -bindWebApp Command Subswitches

Subswitch	Description
`-appName`	Required. The name of the parent application as specified at deployment time.
`-webModuleName`	Required. The name of the Web module to be bound. This should be the name of the WAR file contained within the EAR file, without the `.WAR` extension.
`-webSiteName`	Optional. The name of the *name*`_web-site.xml` file that denotes the Web site to bind the Web module to. If omitted, all modules are bound to `default-web-site`.
`-contextRoot`	Optional. The context root for the Web module. This will be appended to the URL used to access the application through a Web browser; for example: `http://localhost:8888/petstore`. If not supplied, the context root specified in the parent application's `application.xml` deployment descriptor will be used.

Redeploying an Archive

The `-redeploy` command can be used to redeploy a previously-deployed archive.

Note that this operation performs a "graceful" redeployment as it stops the application if it is running, then undeploys the archive. It then deploys and restarts the application. Redeploying an archive with the `-deploy` command, in contrast, does not stop the application—it simply undeploys, redeploys and then restarts it.

The syntax is as follows:

```
java -jar admin_client.jar uri adminId adminPassword -deploy -file
path/filename -deploymentName appName [-keepSettings] [-sequential]
```

Table 6–10 -redeploy Command Subswitches

Subswitch	Description
`-file`	Required. The path and filename of the EAR file to deploy.
`-deploymentName`	Required. The user-defined application deployment name, used to identify the application within OC4J. This value must exactly match the name of the existing application on the server.
`-keepSettings`	Optional. If included, the redeployed application will fetch and use the deployment plan from the previous deployment. Values set in deployment descriptors packaged within the archive will be ignored. If not specified, values will be set to those in the deployment descriptors packaged with the archive.
`-sequential`	Optional. Include to deploy the archive to each OC4J instance within the cluster in sequence. The redeployment on each target must complete before continuing on to the next target. Requests will not be routed to an instance while the EAR is being deployed to it. If not included, the archive is simultaneously deployed to all instances by default. This option is valid in a clustered environment only; it is not valid for standalone OC4J.

Undeploying an Archive

The `-undeploy` command removes an application or standalone Web or connector module from the target OC4J instances, as the following topics describe:

- Undeploying an EAR or Standalone WAR
- Undeploying a Standalone RAR

Undeploying an EAR or Standalone WAR

Undeploying an EAR or standalone Web module removes it from the OC4J runtime. Existing Web site bindings are also deleted.

The syntax is as follows. Note that the name of the application or module must be supplied.

```
java -jar admin_client.jar uri adminId adminPassword -undeploy appName
```

Undeploying a Standalone RAR

Because a standalone RAR is a resource that may be shared by multiple applications, undeploying a standalone RAR requires a restart of the `default` application. This in turn will force a restart of all deployed applications, which are child applications of `default`. See "Starting/Stopping/Restarting an Application" on page 6-14 for instructions on restarting an application.

The syntax is as follows. Note that the `-isConnector` subswitch must be included along with name of the connector.

```
java -jar admin_client.jar uri adminId adminPassword -undeploy
connectorName -isConnector
```

Updating Modified Classes Only in a Deployed EJB Module

The `-application` command includes an `-updateEJBModule` subswitch that allows incremental or partial redeployment of EJB modules within an application running in an OC4J instance. This feature makes it possible to redeploy only those beans within an EJB JAR that have changed, without requiring the entire module to be redeployed.

The syntax is as follows. Note that the name of the application the EJB JAR is part of must be supplied. If updating a standalone EJB module, specify the `default` application.

```
java -jar admin_client.jar uri adminId adminPassword -updateEJBModule
-appName appName -ejbModuleName ejbJarName -file path/ejbJarName
```

For example:

```
java -jar admin_client.jar deployer:oc4j:opmn://node1.company.com/home oc4jadmin
password -updateEJBModule -appName petstore -ejbModuleName customerEjb.jar -file
build/customerEjb.jar
```

Table 6–11 -updateEJBModule Syntax

Option	Description
`-appName`	Required. The name of the application the EJB is part of. If updating a standalone EJB module, specify the `default` application.

Table 6–11 (Cont.) -updateEJBModule Syntax

Option	Description
`-ejbModuleName`	Required. The name of the EJB JAR file to be updated as defined in `application.xml`.
`-file`	Required. The path and file name of the updated EJB JAR.

Creating and Managing Shared Libraries

Shared libraries can be created or removed on OC4J instances.

- Installing a Shared Library
- Modifying an Existing Shared Library
- Viewing the Contents of a Shared Library
- List All Shared Libraries

Note: In the current release, the commands listed below can only be run against a single OC4J instance, either OPMN managed or standalone. The commands will not create or modify shared libraries across a cluster.

Installing a Shared Library

You can use the `-publishSharedLibrary` command to create the shared library directory structure and install the binaries that comprise the library within it on a single OC4J instance. The shared library will be created in the *`ORACLE_HOME`*`/j2ee/`*`instance`*`/shared-lib` directory within the OC4J instance.

The command will also declare the shared library within a *`shared-library`* element in the `server.xml` file on each OC4J instance, making it available to applications.

The syntax is as follows. Note that the path and file name for multiple code sources—binaries that will comprise the shared library—can be specified, each separated from the next by a space.

```
java -jar admin_client.jar uri adminId adminPassword -publishSharedLibrary
-name libName -version libVersion [-parentName parentLibName]
[-parentVersion parentLibVersion] [-installCodeSources path [path ...]]
[-addCodeSources path [path ...]]  [-imports sharedLibName
[:min-version][,max-version] [sharedLibName ...]]
```

The following command deploys the `acme.common:2.5` shared library to a single OC4J instance.

```
java -jar admin_client.jar
deployer:oc4j:opmn://server.company.com:6004/instance2/home
oc4jadmin password -publishSharedLibrary -name acme.common version 2.5
-installCodeSources /myserver/tmp/acme-apis.jar /myserver/tmp/acmeImpl.jar
```

The resulting directory structure within the target OC4J server would be:

```
ORACLE_HOME/j2ee/home/shared-lib
  /acme.common
    /2.5
      acme-apis.jar
      acmeImpl.jar
```

Table 6–12 -publishSharedLibrary Command Subswitches

Subswitch	Description
`-name`	Required. The name of the shared library. Where common APIs are implemented by multiple vendors, the name should include both the vendor name and the name of the technology; for example, `oracle.jdbc` or `xerces.xml`.
`-version`	Required. The shared library version number. This value should ideally reflect the code implementation version.
`-parentName`	Optional. The name of the parent shared library, if applicable.
`-parentVersion`	Optional. The parent shared library version number, if applicable.
`-installCodeSources`	The path and file names for one or more JAR or ZIP files to upload to the OC4J server to add to the shared library. Separate each path/file name string with a space.
`-addCodeSources`	Optional. The path and file names for JAR or ZIP files that have already been uploaded to the OC4J server to add to the shared library. Separate each path/file name string with a space.
`-imports`	Optional. The name of one or more existing shared libraries to import into this shared library. Separate each name string with a space. Note that you can optionally specify the maximum and/or minimum version of the library to import.

Modifying an Existing Shared Library

You can use the `-modifySharedLibrary` command to modify the contents of an existing shared library. The command will also update the shared library definition within the `server.xml` file on each OC4J instance.

The syntax is as follows. Note that the path and file name for multiple code sources—binaries that will comprise the shared library—can be specified, each separated from the next by a space.

```
java -jar admin_client.jar uri adminId adminPassword -modifySharedLibrary
-name libName -version libVersion [-installCodeSources path [path ...]]
[-addCodeSources path [path ...]] [-removeCodeSources path [path ...]]
[-addImports sharedLibName[:min-version][,max-version] [sharedLibName ...]]
[-removeImports sharedLibName[:min-version][,max-version] sharedLibName ...]]
```

The following command updates the `acme.common:2.5` shared library.

```
java -jar admin_client.jar
deployer:oc4j:opmn://server.company.com:6004/instance2/home
oc4jadmin password -modifySharedLibrary -name acme.common version 2.5
-addCodeSources /myserver/tmp/acme-helpers.jar
```

Table 6–13 -modifySharedLibrary Command Subswitches

Subswitch	Description
`-name`	Required. The name of the shared library to update.
`-version`	Required. The version number of the shared library to update.
`-installCodeSources`	Optional. The path and file name to a JAR or ZIP file to be uploaded to the OC4J server and installed as part of the shared library. Separate each path/file name string with a space.

Table 6–13 (Cont.) -modifySharedLibrary Command Subswitches

Subswitch	Description
`-addCodeSources`	Optional. The path and file name for one or more JAR or ZIP files that have already been uploaded to the OC4J server to add to the shared library. Separate each path/file name string with a space.
`-removeCodeSources`	Optional. The path and file name for JAR or ZIP files to remove from the shared library.
`-addImports`	Optional. The name of one or more existing shared libraries to import into this shared library. Separate each name string with a space. Note that you can optionally specify the maximum and/or minimum version of the library to import.
`-removeImports`	Optional. The name of one or more existing shared libraries to remove from this shared library. Note that you can optionally specify the maximum and/or minimum version of the library to remove.

Viewing the Contents of a Shared Library

Use the `-describeSharedLibrary` command to view the code sources and imported shared libraries that comprise the specified shared library.

The syntax is as follows:

```
java -jar admin_client.jar uri adminId adminPassword -describeSharedLibrary
-name libName -version libVersion
```

List All Shared Libraries

Use the `-listSharedLibraries` command to output a list of all shared libraries defined on the target OC4J instance.

The syntax is as follows:

```
java -jar admin_client.jar uri adminId adminPassword -listSharedLibraries
```

Starting/Stopping/Restarting an Application

You can use `admin_client.jar` to start, stop or restart an application and its child applications on a specific OC4J instance or across an entire cluster. If a file within the application has been modified, the application will be automatically redeployed at startup.

Note that you can even stop and start Application Server Control Console (`ascontrol`) using these commands.

The syntax is as follows:

```
java -jar admin_client.jar uri adminId adminPassword -start|-stop appName
```

The following example stops the `ascontrol` application on `node2` within the cluster:

```
java -jar admin_client.jar deployer:oc4j:opmn://node2.company.com:6004/home
oc4jadmin password -stop ascontrol
```

7

Using the admin.jar Utility

OC4J provides a command-line utility called `admin.jar` that can be used to perform operations on an active OC4J instance in a standalone OC4J installation. Among other things, you can use this utility to restart and stop OC4J, deploy applications and gather information on current resource usage.

This chapter includes the following topics:

- Overview of admin.jar Usage
- Managing a Standalone OC4J Instance
- Deploying/Undeploying Applications
- Managing Applications
- Managing Web Sites
- Managing Data Sources
- Deploying/Undeploying Connectors

Overview of admin.jar Usage

The `admin.jar` utility is installed by default in `ORACLE_HOME/j2ee/home` in a standalone OC4J instance.

Note that OC4J must be started before this utility can be used. Also note that the utility cannot be used to start OC4J, although it can be used to stop and then restart an instance.

This section covers the following topics:

- Understanding the admin.jar Syntax
- Printing Help to the Console

Note: The `admin.jar` utility can be used only to manage a single OC4J instance in a standalone OC4J installation.

Use Oracle Process Manager and Notification Server (OPMN) to manage OC4J instances running as components of Oracle Application Server.

Due to its more advanced capabilities, the `admin_client.jar` utility should be used instead of `admin.jar`. See Chapter 6, "Using the admin_client.jar Utility" for details on using this utility.

Understanding the admin.jar Syntax

The `admin.jar` utility uses the following syntax. The parameters are described in Table 7–1.

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
   adminPassword options
```

As an example, the following command will force a graceful shutdown of the OC4J server. The value supplied for *`oc4jOrmiPort`* is the default, `23791`. The user name supplied for *`adminId`* is the user name for the default administrator account, `oc4jadmin`.

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -shutdown
```

Note that some of these commands include an `-application` switch that takes the name of the application to affect. The value is the name of the specific application to affect, as defined within the appropriate `<application>` element in the `server.xml` configuration file.

Table 7–1 Setting the Host and Login Information

Parameter	Description
`oc4jHost``:`*`oc4jOrmiPort`*	The host name and port of the OC4J server on which you are invoking `admin.jar`. The `admin.jar` tool uses the OC4J Remote Method Invocation (ORMI) protocol to communicate with the OC4J server. Therefore, the host name and port identified by these variables are defined in the `rmi.xml` file for the OC4J server to which you are directing the request. The OC4J default port for the ORMI protocol is `23791`. This value can be omitted if not changed. Configure both the host name and port number - if not using the default - in the `rmi.xml` file in the `<rmi-server>` element, as follows: `<rmi-server port="`*`oc4jOrmiPort`*`" host="`*`oc4jHost`*`" />`
`adminId` *`adminPassword`*	The OC4J administration user name and password. The user name for the default administrator account is `oc4jadmin`.

Printing Help to the Console

To print the online help text for the `admin.jar` commands to the console, simply type `-help` after *`oc4jHost`*`:`*`oc4jOrmiPort adminId adminPassword`*. For example:

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -help
```

Managing a Standalone OC4J Instance

This section outlines the functionality provided by `admin.jar` for managing an OC4J server. It includes the following sections:

- Stopping and Restarting OC4J in a Standalone Environment
- Forcing OC4J to Check for Modified Files

Stopping and Restarting OC4J in a Standalone Environment

You can use `admin.jar` to shut down a standalone instance of the OC4J server and then restart it.

The following command forces a shutdown of the OC4J server, which terminates all threads immediately. The string entered as the `reason` for the shutdown is written to the server log file, `ORACLE_HOME/j2ee/home/log/server.log`.

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -shutdown force
  need_to_reboot_host_machine
```

Table 7–2 Options for OC4J Server Shutdown/Restart

Option	Description
`-shutdown`	Shuts down the OC4J server. `[ordinary|force]`: The type of shutdown. The default is `ordinary`, which allows each thread to terminate normally. The `force` option terminates all threads immediately. `[reason]`: You can optionally specify a reason for the shutdown as a string that is written to the `ORACLE_HOME/j2ee/home/log/server.log` file. Spaces are not allowed in the string.
`-restart`	Restarts the OC4J server. The container must have been started with `oc4j.jar`. `[reason]`: You can optionally specify a reason for the restart as a string that is written to the `ORACLE_HOME/j2ee/home/log/server.log` file. Spaces are not allowed in the string.
`-version`	Prints the installed version of OC4J to the console, then exits.

Forcing OC4J to Check for Modified Files

You can force OC4J to check the server directory structure for modified files and reload any that have changed, using the `-updateConfig` option.

Note that the value of the `checkForUpdates` flag must be set to either `all` or `adminClientOnly` (the default setting) to use this feature. See *Oracle Containers for J2EE Deployment Guide* for details on the `checkForUpdates` flag.

Table 7–3 Option for Checking for Updated Files

Option	Description
`-updateConfig`	Forces OC4J to check files for changes and reload any files that have been modified.

Deploying/Undeploying Applications

You can use `admin.jar` to deploy or undeploy J2EE applications to or from a standalone OC4J instance.

Usage Notes:

- `admin.jar` cannot be used to deploy applications to an OPMN-managed OC4J instance.
- `admin.jar` supports deployment of EAR files only. It does not allow deployment of standalone modules, such as a Web module packaged in a WAR file.
- `admin.jar` does not accept a deployment plan. Any archive deployed using this utility must include the required OC4J-specific deployment descriptor files, such as `orion-application.xml` or `orion-web.xml`.

Deploying an application is a two-step process: You must first deploy the archive into OC4J, then bind the Web module to the Web site that will be used to access the application.

The `-deploy` command is first used to deploy the application:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
   adminPassword -deploy -file path/filename
    -deploymentName appName -targetPath deploy_dir
```

Once the archive is deployed, the `-bindWebApp` command is used to bind a Web application to the Web site it will be accessed through:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId    adminPassword
-bindWebApp appName webAppName
   webSiteName contextRoot
```

For example, the following command deploys the `utility` application into OC4J:

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -deploy -file
utility.ear -deploymentName utility
```

Next, the following example binds the `utility` application and its `utility-web` Web module to the `default` OC4J Web site:

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -bindwebapp utility
utility-web default-web-site /utility
```

Table 7–4 Options for Application Deployment

Option	Description
`-deploy`	Deploys an application. Supply relevant information using the following subswitches: `-file` *`filename`*: Required. The path and filename of the EAR file to deploy. `-deploymentName` *`appName`*: Required. The user-defined application deployment name. This same name is used to identify the application within OC4J. It is also provided when you want to undeploy the application. `-targetPath` *`path`*: Optional. The path on the server node to deploy the EAR to. If not specified, the EAR is deployed to the *`ORACLE_HOME`*`/j2ee/`*`instance`*`/applications/` directory by default. The deployed EAR file is also copied to this directory. Each successive deployment will cause this EAR file to be overwritten. `-parent` *`appName`*: Optional. The parent application of this application. When deployed, any method within the child application can invoke any method within the parent application. In no parent is specified, the `default` application serves as the default parent. `-deploymentDirectory` *`path`*: Optional. The directory containing the OC4J-specific deployment descriptors and generated files, such as compiled JSP classes and EJB wrapper classes. The default directory is *`ORACLE_HOME`*`/j2ee/`*`instance`*`/application-deployments/`. `-iiopClientJar` *`path/filename`*: Optional. Include to generate IIOP stubs for the home, remote and local interfaces packaged within each EJB JAR included in the EAR. You can optionally specify the path and filename of the JAR to output the generated stubs to. Otherwise, copies of the stubs will be output to an archive named `_iiopClient.jar` in a new subdirectory with the same name as the deployed EJB JAR in *`ORACLE_HOME`*`/j2ee/home`*`app-name`*`/application-deployments/`. Note that the `GenerateIIOP` system property must be enabled at OC4J startup to use this feature. For example: `java -DGenerateIIOP=true -jar oc4j.jar`

Table 7–4 (Cont.) Options for Application Deployment

Option	Description
`-bindWebApp`	Binds a Web application to the specified Web site and root. ■ *appName*: The application name, which is the same name set as the value for `-deploymentName` in the `-deploy` option. ■ *webAppName*: The name of the Web module. This should be the name of the WAR file contained within the EAR file, without the `.WAR` extension. ■ *webSiteName*: The name of the *name*`_web-site.xml` file that denotes the Web site that this Web application should be bound to. ■ *contextRoot*: The root context for the Web module. This will be appended to the URL used to access the application through a Web browser; for example `http://localhost:8888/utility`. This option creates an entry in the *name*`-web-site.xml` configuration file that was denoted in the *web_site_name* variable.
`-undeploy` *appName*	Removes the deployed J2EE application from the OC4J instance. The value of *appName* is the name of the application within OC4J, as defined in an *application* element within `ORACLE_HOME/j2ee/home/config/server.xml`. Undeploying an application results in the following: ■ The application is removed from the OC4J runtime and the `server.xml` file. ■ Bindings for all the application's Web modules are removed from all the Web sites to which the Web modules were bound. ■ Application files are removed from both the `applications` and `application-deployments` directories. `-keepFiles`: This optional subswitch is deprecated in OC4J 10*g* (10.1.3).

Managing Applications

This section outlines the functionality provided by `admin.jar` for managing applications in a standalone OC4J instance. It includes the following sections:

- Starting/Stopping/Restarting an Application
- Updating an EJB Module Within an Application

Starting/Stopping/Restarting an Application

You can use `admin.jar` to start, stop and restart an application that has been stopped in a standalone OC4J instance.

The following example restarts a specific application running on OC4J. If a file within the application has been modified, the application or module will be automatically redeployed.

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -application
   myapplication -restart
```

Table 7–5 Options for Application Restart

Option	Description
`-application` *appName* `-start`	Starts the specified application and any child applications.
`-application` *appName* `-stop`	Stops the specified application and any child applications.
`-application` *appName* `-restart`	Restarts the specified application and any child applications. If OC4J polling is enabled and a file within the application has been modified, the application will be redeployed.

Updating an EJB Module Within an Application

The `admin.jar` utility includes an `-updateEJBModule` option that allows incremental or partial redeployment of EJB modules within an application running in an OC4J instance. This option is primarily intended to be used by an application developer to redeploy the JAR file directly from his/her development environment.

The syntax is as follows:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
    adminPassword -application appName -updateEJBModule relativePath
    [-file path/ejbJarName]
```

For example, the following commands can be used to update the `customerEjb.jar` module of the `petstore` application. Assume the following directory structure on the developer's machine:

```
/work
  /src    - application source code
  /build  - compiled class files
  /dist   - assembled EAR and JAR files
```

If the updated EJB JAR is in the `/dist` directory, in a location matching the relative path defined in the application's `application.xml` J2EE standard deployment descriptor, the following command could be issued from the `/dist` directory:

```
java -jar $OC4J_HOME/admin.jar ormi://myoc4jserver:23791 oc4jadmin password
    -application petstore -updateEJBModule customerEjb.jar
```

If the updated file is located within the `/build` directory, the following command specifying the JAR location in the `-file` option can be issued from the `/dist` directory:

```
java -jar admin.jar ormi://myoc4jserver:23791 oc4jadmin password
    -application petstore -updateEJBModule customerEjb.jar
    -file build/customerEjb.jar
```

Table 7–6 Options for Updating an EJB Module

Option	Description
`-application` *appName* `-updateEJBModule`	Updates the specified EJB module with new EJBs. ■ *relativePath*: The relative path to the EJB JAR containing the updated beans as defined in the application's `application.xml` J2EE deployment descriptor. ■ `-file` *path*: The path and file name of the updated EJB JAR if the file's location does not match the relative path specified in the `application.xml` deployment descriptor.

Managing Web Sites

The `-site` option enables you to configure new Web sites, including secure sites, for use by applications deployed into OC4J. You can also retrieve a list of existing sites; test existing sites; and update or remove existing sites.

The syntax of the `-add` option, which configures a new site, is as follows:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
   adminPassword -site -add [-host hostName] -port port -display-name name
[-virtual-hosts hostNames] [-secure [true|false]] [-factory class] [-keystore
path] [-storepass password] [-provider class] [-needs-client-auth [true|false]]
```

For example, the following command structure configures a new Web site on port `8899` with two virtual hosts:

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -site -add -host
www1.acme.com -port 8899 -display-name MyServer -virtual-hosts
MyServer.com,MyServer2.com
```

The next example configures a secure Web site to receive HTTPS requests on port `4443`. See "Configuring a Secure Web Site in OC4J" on page 13-9 for instructions on creating secure Web sites.

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -site -add
-host www1.acme.com -port 8899 -display-name MySecureSite -secure
true -factory com.evermind.ssl.JSSESSLServerSocketFactorykeystore
-keystore ../../server.keystore -storepass password
-provider com.sun.net.ssl.internal.ssl.Provider -needs-client-auth true
```

Table 7–7 Options for Web Site Administration

-site options	Description
`-site -add`	Installs a new Web site. Supply information with the following subswitches: `-host` *`hostName`*: The name of the host or the IP address hosting the Web site. Optional. `-port` *`port`*: The Web site port. Required. `-display-name` *`name`*: A user-friendly "displayable" name of the Web site. `-virtual-hosts` *`hostNames`*: One or more virtual Web sites sharing the same IP address. The value is a comma-delimited list of host names tied to this Web site. Optional. `-secure [true|false]`: The value is `true` if the Web site is secure. The default is `false`. `-factory` *`className`*: The name of the class extending `SSLServerSocketFactory` if you are not using Java Secure Socket Extension (JSSE). Optional. `-keystore` *`path`*: The relative or absolute path to a keystore. Optional. `-storepass` *`password`*: The keystore password. Optional. `-provider` *`provider`*: The security provider to use if not using JSSE. If not specified, the Sun Microsystems implementation - `com.sun.net.ssl.internal.ssl.Provider` - is used by default. The JSSE defines a provider interface that other security providers can implement. Optional. `-needs-client-auth [true|false]`: If set to `true`, a client that wants to access the Web site must identify itself with a digital certificate. The default is `false`.
`-site -remove`	Removes an existing Web site. Supply the host and port of this Web site in the following subswitches: `-host` *`hostName`*: The Web site host to be removed. `-port` *`port`*: The Web site port to be removed.
`-site -test`	Tests an existing Web site. Supply the host and port of the Web site to be tested, in the following subswitches: `-host` *`hostName`*: The Web site host to be tested. `-port` *`port`*: The Web site port to be tested.
`-site -list`	Lists all existing Web sites configured within the OC4J instance.

Table 7–7 (Cont.) Options for Web Site Administration

-site options	Description
`-site -update`	Updates an existing Web site. Supply information with the following subswitches: `-oldHost` *`hostName`*: The name of the host or the IP address of the current Web site host. `-oldPort` *`port`*: The current port. `-newHost` *`hostName`*: The hostName or IP address of the new Web site host. `-newPort` *`port`*: The new port. `-display-name` *`name`*: A user-friendly "displayable" name of the Web site. `-virtual-hosts` *`hostNames`*: One or more virtual Web sites sharing the same IP address. The value is a comma-delimited list of host names tied to this Web site. Optional. `-secure [true|false]`: The value is `true` if the Web site is secure. The default is `false`. `-factory` *`className`*: The name of the class extending `SSLServerSocketFactory` if you are not using Java Secure Socket Extension (JSSE). Optional. `-keystore` *`path`*: The relative or absolute path to a keystore. Optional. `-storepass` *`password`*: The keystore password. Optional. `-provider` *`provider`*: The security provider to use if not using JSSE. If not specified, the Sun Microsystems implementation - `com.sun.net.ssl.internal.ssl.Provider` - is used by default. The JSSE defines a provider interface that other security providers can implement. Optional. `-needs-client-auth [true|false]`: If set to `true`, a client that wants to access the Web site must identify itself with a digital certificate. The default is `false`.

Managing Data Sources

Use `admin.jar` to create, remove, list or test data sources for a specific application. You can also convert a pre-Oracle Application Server 10*g* Release 3 (10.1.3) `data-sources.xml` file to the new file format.

Creating an Application-Specific Data Source

The syntax of the `-installDataSource` option, which configures a new application-specific data source, is as follows:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-application appName -installDataSource -jar path
-url url -location jndiName [-pooledLocation jndiName]
[-xaLocation jndiName] [-ejbLocation jndiName] -username name
-password password [-connectionDriver className] -className className
[-sourceLocation jndiName][-xaSourceLocation jndiName]
```

An example follows.

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -application myapp
-installDataSource -jar C:/jdbc/lib/ojdbc14dms.jar
-url jdbc:oracle:thin:@dev2:1521:main -location jdbc/OracleUddi
-username dbuser -password dbpw -className oracle.jdbc.pool.OracleDataSource
```

Table 7–8 Options for Data Source Management

Option	Description
`-application` *appName* `-installDataSource`	Installs a new data source for the specified application. Supply data source information within the following subswitches: `-jar` *path*: Required. The path to the JAR file containing the JDBC driver that is to be added to the OC4J server. `-url` *url*: Required. The JDBC database URL. `-location` *jndiName*: Required. The JNDI name for the raw data source. For example, `"jdbc/DefaultPooledDS"`. `-pooledLocation` *jndiName*: Optional. The JNDI name for the pooled data source. For example, `"jdbc/DefaultPooledDS"`. `-xaLocation` *jndiName*: Optional. The JNDI name for the XA source. For example, `"jdbc/xa/DefaultXADS"`. Required if `-ejbLocation` is specified. `-ejbLocation` *jndiName*: Optional. The JNDI name for the container-managed transactional data source. This is the only data source that can perform global JTA transactions. For example, `"jdbc/DefaultDS"`. `-username` *name*: Required. The user name to log in to the database. `-password` *password*: Required. The password to log in to the database. `-connectionDriver` *className*: Optional. The JDBC database driver class. `-className` *className* Required. The data source class name, such as `com.evermind.sql.DriverManagerDataSource`. `-sourceLocation` *jndiName*: Optional. The JNDI name of the underlying data source of this specialized data source. `-xaSourceLocation` *jndiName*: Optional. The JNDI name of the underlying XA data source of this specialized data source.

Listing/Testing/Removing Existing Data Sources

You can use `admin.jar` to list, test or even delete data sources tied to a specific application.

Table 7–9 Options for Application and Data Source Management

Option	Description
`-application` *appName* `-listDataSource`	Retrieves the statically configured information about each installed data source object.

Table 7–9 (Cont.) Options for Application and Data Source Management

Option	Description
`-application` *appName* `-testDataSource`	Tests an existing data source. Supply information with the following subswitches: `-location` *jndiName*: The namespace location for the data source. For example, `jdbc/DefaultDS`. Required. `-username` *name*: The user name you use to log in along with a password. Optional. `-password` *password*: The password to log in with. Optional.
`-application` *appName* `-removeDataSource`	Removes an existing data source. Supply information with the following subswitch: `-location` *jndiName*: The namespace location for the data source. For example, `jdbc/DefaultDS`. Required.

Converting Existing Data Sources to the New Configuration

The `-convertDataSourceConfiguration` option converts a pre-Release 3 (10.1.3) `data-sources.xml` file to the new file format.

The syntax is as follows:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
   adminPassword -convertDataSourceConfiguration legacyFileName convertedFileName
```

For example, the following command converts an existing configuration and writes it to a new file:

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password
-convertDataSourceConfiguration C:\oc4j\j2ee\home\config\data-sources.xml
C:\new\data-sources.xml
```

Ideally, you should rename the "old" `data-sources.xml` after the conversion, rather than delete it, as it contains information that may be needed for reference. After the "new" file has been generated, copy it into the directory containing the legacy file.

Note that the generated `data-sources.xml` file may include JNDI entries that are not needed but are transferred from the legacy file. Edit the file to remove these entries.

Table 7–10 Options for Data Source File Conversion

Arguments	Description
legacyFileName	The fully qualified path to the "old" `data-sources.xml` file you want to convert.
convertedFileName	The fully qualified path to the "new" `data-sources.xml` file containing the converted configuration.

Deploying/Undeploying Connectors

You can use one of the following commands to deploy or undeploy a Java Connector Architecture-compliant resource adapter packaged in a RAR file.

Table 7–11 Options for Application Deployment

Option	Description
`-deployconnector`	Deploys a connector. Supply application information in the following subswitches: `-file` *`path`*: Required. The path and filename of the RAR file to deploy. `-name` *`name`*: The name of the resource adapter. `-nativeLibPath` *`path`*: The path to the directory containing native libraries (such as DLLs) within the RAR file. `-grantAllPermissions`: Include to grant all runtime permissions requested by the resource adapter, if required.
`-undeployconnector`	Undeploys the specified connector. `name` *`name`*: The name of the connector to undeploy. Note that if you are undeploying a standalone RAR, the `default` application must be restarted.

8

Configuring and Managing Clusters

This chapter explains how to configure and manage cluster topologies in an Oracle Application Server environment. It includes the following topics:

- Clustering Overview
- Configuring a Cluster
- Viewing the Status of a Cluster
- Load Balancing with Oracle HTTP Server
- Configuring Application Mount Points
- Replicating Changes Across a Cluster
- Creating and Managing Additional OC4J Instances

Note that application clustering - the clustering of applications deployed to Oracle Application Server nodes for the purpose of session or state replication - is covered in Chapter 9, "Application Clustering in OC4J".

Clustering Overview

This section provides an overview of the clustering mechanisms supported in Oracle Application Server 10*g* Release 3 (10.1.3), and notes the significant changes in functionality between this current release and previous releases. It includes the following topics:

- How Clustering Works
- Supported Clustering Models
- Changes in Clustering

How Clustering Works

In the current release, a cluster topology is defined as two or more loosely connected Oracle Application Server nodes.

The connectivity provided within a cluster is a function of Oracle Notification Server (ONS), which manages communications between Oracle Application Server components, including OC4J and OHS. The ONS server is a component of Oracle Process Manager and Notification Server (OPMN), which is installed by default on every Oracle Application Server host. When configuring a cluster topology, you are actually connecting the ONS servers running on each Oracle Application Server node.

Previous releases of Oracle Application Server supported clustering of a fully connected set of server nodes only, which meant that each node had to be explicitly

specified in the ONS configuration file (`ons.conf`). When a node was added or removed from the cluster, the configuration had to be updated on each server node, and the server restarted.

The current release supports a new "dynamic discovery" mechanism, enabling the cluster to essentially manage itself. In this framework, each ONS maintains a map of the current cluster topology. When a new ONS is added to the cluster, each existing ONS adds the new node and its connection information to its map. At the same time, the new ONS adds all of the existing nodes to its map. Alternatively, when an ONS is removed from the cluster, the maps for the remaining nodes are updated with this change.

As of Oracle Application Server Release 3 (10.1.3), the ONS configuration file (`ons.conf`) is no longer used. Instead, ONS configuration data is set in the `<notification-server>` element within `opmn.xml`, the OPMN configuration file located in the *`ORACLE_HOME`*`/opmn/conf` directory on each node. Clustering configuration in turn is set within a `<topology>` subelement. Note that only one `<topology>` subelement is allowed.

The example below illustrates a cluster topology configuration in `opmn.xml`:

```
<notification-server>
  <topology>
    <discover list="*225.0.0.20:8001"/>
  </topology>
  ...
</notification-server>
```

The clustering configuration specified in the `<topology>` element applies to all instances of Oracle Application Server components - including OHS and OC4J - installed on the node. Note that all nodes within a cluster topology must have the same configuration specified in the `opmn.xml` file.

Supported Clustering Models

The following clustering models are supported:

- Dynamic node discovery

  In this configuration, each ONS node within the same subnet announces its presence with a multicast message. The cluster topology map for each node is automatically updated as nodes are added or removed, enabling the cluster to be self-managing.

  See "Configuring Dynamic Node Discovery Using Multicast" on page 8-5 for configuration instructions.

- 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 via this server. Hubs in one topology can be connected to those in another.

  See "Configuring Static Discovery Servers" on page 8-11.

- Connection of isolated topologies via gateways

  This configuration is used to connect topologies separated by firewalls or on different subnets using specified "gateway" nodes.

  See "Configuring Cross-Topology Gateways" on page 8-14 for details.

- Manual node configuration

  In this 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 10*g* Release 3 (10.1.2) and is supported primarily to provide backward compatibility.

  See "Configuring Static Node-to-Node Communication" on page 8-17 for instructions.

Changes in Clustering

The following are changes in cluster configuration in Oracle Application Server 10*g* Release 3 (10.1.3) from previous releases.

- The Distributed Configuration Management (DCM) framework, used in prior releases of Oracle Application Server to replicate common configuration information across a cluster, is not included in the current release. This means that:
  - Configuration using the `dcmctl` command line utility or Application Server Control Console is no longer supported.
  - Cluster configurations must now be manually replicated in the `opmn.xml` file installed on each node within the cluster.
- The ONS configuration file (`ons.conf`) is no longer used. ONS connection data is now set in the `<notification-server>` element within `opmn.xml`, the OPMN configuration file located in the `ORACLE_HOME/opmn/conf` directory on each node containing an OC4J or OHS instance.
- Each node is no longer required to be manually configured to connect to every other node in the cluster.

Configuring a Cluster

This section contains instructions on configuring the following clustering models:

- Configuring Dynamic Node Discovery Using Multicast
- Configuring Static Discovery Servers
- Configuring Cross-Topology Gateways
- Configuring Static Node-to-Node Communication

Configuring Dynamic Node Discovery Using Multicast

Dynamic node discovery is the most straightforward clustering configuration. In this model, each ONS node broadcasts a simple multicast message announcing its presence, enabling nodes within the cluster to dynamically "discover" one another.

The following tools can be used to add OC4J instances to a cluster using multicast discovery:

- `opmnassociate`

  This utility provides a one-step solution for adding an OC4J instance to a cluster. See "Configuring Multicast Discovery with opmnassociate" on page 8-8 for details.

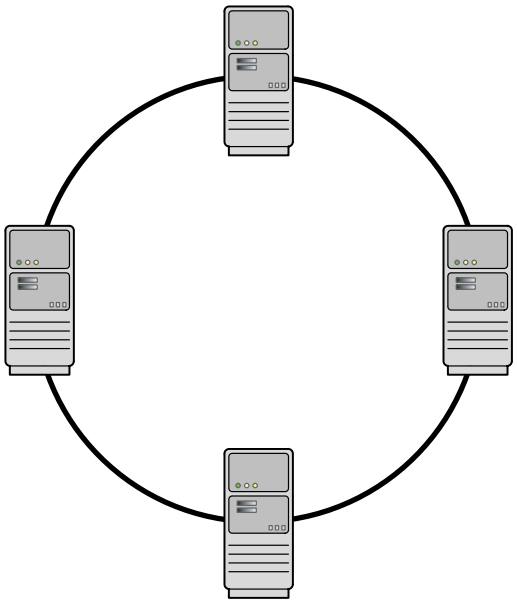
- `opmnctl`

  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. See "Configuring Multicast Discovery with opmnctl" on page 8-9 for details.

Each ONS maintains its own map of the cluster topology. When a new ONS is added to the cluster, each existing ONS adds the new node and its connection information to its map. At the same time, the new ONS adds all of the existing nodes to its map. Alternatively, when an ONS is removed from the cluster, the maps for the remaining nodes are updated with this change.

Figure 8–1 Dynamic Discovery Model



Because multicast messages may be restricted by different network configurations dynamic node discovery may be an option only for ONS nodes that are on the same subnet. However, multiple subnets using dynamic node discovery may be connected using gateway servers. See "Configuring Cross-Topology Gateways" on page 8-14 for details.

Notes:

- All nodes within the topology must be configured to use the same multicast address and port.
- The multicast address must be within the valid address range, which is `224.0.0.1` to `239.255.255.255`.

  Ideally, multicast address and port assignments should be managed by your systems administration staff to avoid potential conflicts with other applications.

The dynamic discovery configuration is set within a `<discover>` subelement of the `<topology>` element in the `opmn.xml` file on each Oracle Application Server instance in the topology. To add a new node to the cluster, simply add this element to its `opmn.xml` file. To remove a node from the cluster, remove this element.

Set the multicast IP address and port as the value for the `list` attribute. Note the asterisk (`*`) preceding the IP address. This character is critical, as it informs OPMN that

the value specified is a multicast address. Multiple values can be specified, each separated from the next by a comma.

```
<opmn>
 <notification-server>
  <port ...  />
  <ssl ... />
  <topology>
    <discover list="*225.0.0.20:8001"/>
  </topology>
 </notification-server>
 ...
</opmn>
```

Note: The `opmn.xml` file must be reloaded for changes made to take effect. Run the following command on the affected node to reload `opmn.xml`:

```
opmnctl reload
```

Note that this command will not affect OPMN-managed components, including OHS, OC4J and deployed applications.

Configuring Multicast Discovery with opmnassociate

The `opmnassociate` utility provides a solution for adding an OC4J instance to a cluster using multicast discovery. It performs the following steps:

- Inserts or updates the `<discover>` element in `opmn.xml` with the specified multicast address and port
- Configures the `default` Web site to receive and respond to requests from Oracle HTTP Server using the Apache JServ Protocol (AJP) by modifying the corresponding `<port>` element in `opmn.xml`
- Restarts OPMN to load the new configuration into the runtime

The `opmnassociate` tool is installed in the *`ORACLE_HOME`*`/bin` directory on each OC4J instance. The tool must be run individually on each instance, and will update only the `opmn.xml` file on that instance.

The syntax is as follows:

```
opmnassociate "*multicastAddress:multicastPort" [-restart]
```

For example:

```
opmnassociate "*225.0.0.20:8001" -restart
```

The asterisk (`*`) preceding the IP address is required.

Note that this tool can only be used to add the default `home` OC4J instance to a cluster; to add other OC4J instances—for example, `home2`—use `opmnctl` as outlined below.

Configuring Multicast Discovery with opmnctl

The OPMN command-line tool, `opmnctl`, supports a new `config topology` command that allows you to specify, update or delete the multicast `<discovery>` entry within `opmn.xml`.

The opmnctl tool is installed in the ORACLE_HOME/opmn/bin directory on each node. The tool must be run individually on each node and will update only the opmn.xml file on that node.

Note for Adding OPMN-Managed Standalone OC4J Instances:

The default Web site in an OPMN-managed OC4J instance that does not include Oracle HTTP Server (J2EE Server and Process Management install type) is configured to listen for HTTP requests by default.

When adding the instance to a cluster, you must configure the Web site to use the Apache JServ Protocol (AJP). This modification is necessary to enable the OC4J instance to receive and respond to requests from Oracle HTTP Server.

Ideally, you should use the opmnctl config port update command to modify the default Web site configuration defined in opmn.xml. See "Configuring Web Sites with opmnctl" on page 13-4 for details.

Inserting or Updating Discovery Data

The update command inserts or updates the <discover> element with the specified values. The syntax is as follows:

```
opmnctl config topology update discover="*multicastAddress:multicastPort"
```

For example:

```
opmnctl config topology update discover="*225.0.0.20:8001"

opmnctl reload
```

Deleting Discovery Data

The delete command removes the <discover> element from opmn.xml, effectively removing the node from the cluster. If the <topology> element contains no other subelements, it will be removed as well.

```
opmnctl config topology delete discover

opmnctl reload
```

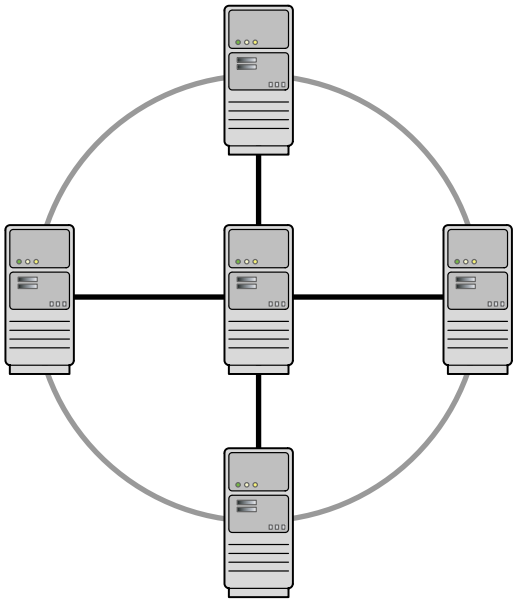
Configuring Static Discovery Servers

This configuration is similar to a peer-to-peer clustering model, with one or more ONS nodes within the same cluster configured to serve as static hubs, or "discovery servers."

Each ONS node in the cluster establishes a connection with a discovery server, which maintains the topology map for the cluster. The discovery server provides the connecting node with the current topology map, enabling the connecting node to communicate with the other ONS nodes within the cluster.

Note that you can use opmnctl to configure the connection to a static discovery server. See "Configuring a Static Discovery Server Connection with opmnctl" on page 8-13 for details.

Figure 8–2 Static Discovery Server Model



Set the TCP/IP connection information for the discovery server within the `<discover>` element in the `opmn.xml` file on each static hub node within the cluster. For example:

```
<opmn>
 <notification-server>
  <port ...  />
  <ssl ... />
  <topology>
   <discover list="node1.company.com:6200"/>
  </topology>
 </notification-server>
 ...
</opmn>
```

The required information is as follows:

- The host name or IP address of the static discovery server
- The OPMN `remote` port, which is defined in the `<port>` element within the `opmn.xml` file installed on the static server, as illustrated below.

  ```
  <port local="6100" remote="6200" request="6003"/>
  ```

Note: The `opmn.xml` file must be reloaded for changes to take effect in the OPMN runtime. Run the following command on the affected node to reload `opmn.xml`:

```
opmnctl reload
```

Note that this command will not affect OPMN-managed components, including OHS, OC4J and deployed applications.

Configuring a Static Discovery Server Connection with opmnctl

The OPMN command line tool, `opmnctl`, supports a new `config topology` command which allows you to specify, update or delete the `<discovery>` entry within `opmn.xml`.

The `opmnctl` tool is installed in the `ORACLE_HOME/opmn/bin` directory on each node. The tool must be run individually on each node, and will only update the `opmn.xml` file on that node.

Inserting or Updating Discovery Data

The `update` command inserts or updates the `<discover>` element with the specified values. The syntax is as follows:

```
opmnctl config topology update discover="serverHost:opmnRemotePort"
```

For example:

```
opmnctl config topology update discover="node.company.com:6200"

opmnctl reload
```

Deleting Discovery Data

The `delete` command removes the `<discover>` element from `opmn.xml`, effectively removing the node from the cluster. If the `<topology>` element contains no other subelements, it will be removed as well.

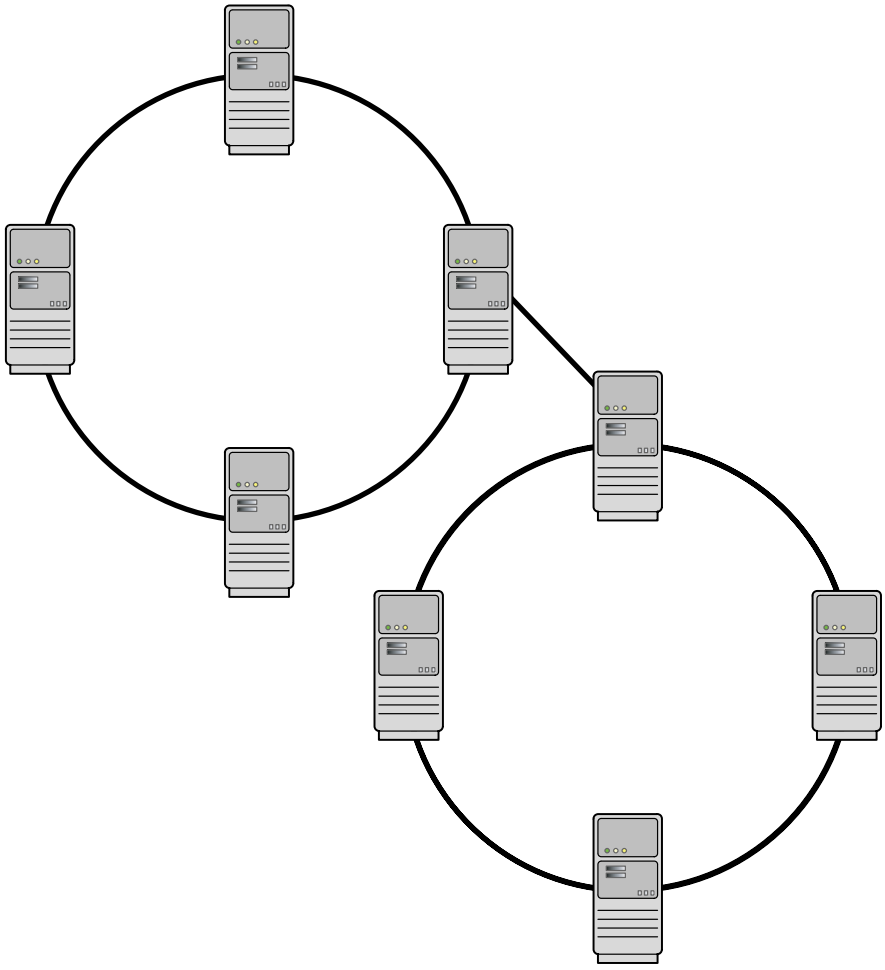
```
opmnctl config topology delete discover

opmnctl reload
```

Configuring Cross-Topology Gateways

For situations in which cluster topologies are on different subnets or are isolated by firewalls or physical locations, specific ONS nodes can be configured as "gateways", enabling ONS notifications to be sent across the disparate topologies.

Figure 8–3 Using Gateway Servers to Connect Topologies



In this model, an ONS node within each isolated topology is configured as a gateway server, which serves as an entry point into the cluster. The gateway configuration is specified within a `<gateway>` subelement of the `<topology>` element.

Set the host and port for the source gateway node and each target node it will connect to as the value for the `list` attribute. The order in which the nodes are listed does not matter.

- For each node, specify the host name or IP address of the server and the OPMN `remote` port, which is defined in the `<port>` element within the `opmn.xml` file installed on the static server, as illustrated below.

  ```
  <port local="6100" remote="6200" request="6003"/>
  ```

- Separate the data for each node with an ampersand (`&`), which must be specified as `&`.
- Include a `/` at the end of the list of nodes.

The example below shows the `opmn.xml` configuration for `node1`, which will connect with gateway nodes `node1` and `node3`. This same configuration can be set on each of these gateway nodes. Note the `/` at the end of the list:

```
<opmn>
 <notification-server>
```

```
  <port ...  />
  <ssl ... />
  <topology>
   <gateway list="node1.com:6201&node2.com:6202&amp;node3.com:6203/"/>
   <discover list="*224.0.0.37:8205"/>
  </topology>
 </notification-server>
 ...
</opmn>
```

Note that in addition to the <gateway> element, the <topology> element includes the <discover> element, which contains the multicast address and port used for dynamic discovery within the node's own cluster.

Alternatively, the entire <topology> element in the preceding example can be copied to the opmn.xml file on every node within the cluster topology. Only node1 will utilize the <gateway> configuration; it will be ignored by the other nodes.

To simplify configuration, you can set the connection data for all gateway nodes - sources and targets - in the <gateway> subelement and then copy this element to the opmn.xml file on each gateway node. Again, the order of the nodes does not matter; each node will simply ignore its own entry in the list.

Note: The opmn.xml file must be reloaded for changes to take effect in the OPMN runtime. Run the following command on the affected node to reload opmn.xml:

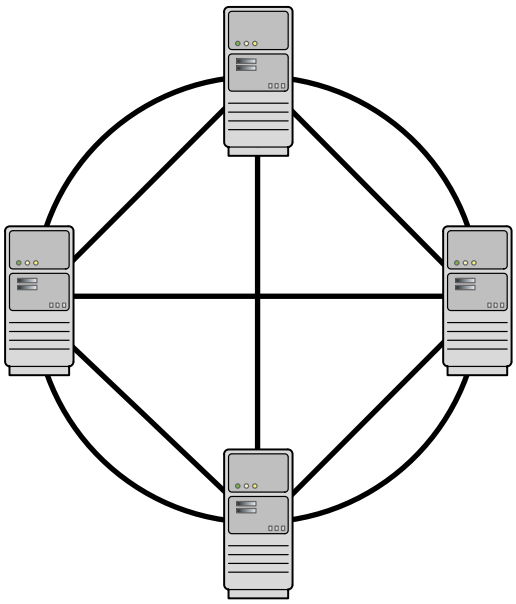
```
opmnctl reload
```

Note that this command will not affect OPMN-managed components, including OHS, OC4J and deployed applications.

Configuring Static Node-to-Node Communication

The static configuration model is essentially the same mechanism used in Oracle Application Server 10.1.2 and 9.0.4. It continues to be supported primarily to provide backward compatibility with these earlier releases.

Figure 8–4 Static Node-to-Node Model



In this configuration, a "node list" containing the host address and ONS remote listener port for each node in the cluster is supplied. Note that prior to Oracle Application Server 10.1.3, when ONS configuration data was integrated into `opmn.xml`, this configuration would have been set in the `ons.conf` configuration file.

Define the host address and the ONS remote listener port - specified within the `<port>` subelement of `<notification-server>` - for each node in the cluster within the `<nodes>` subelement. Separate each node from the next with a comma.

For example:

```
<opmn>
 <notification-server>
  <port local="6101" remote="6202" request="6004"/>
  <ssl ... />
  <topology>
   <nodes list="node1-sun:6201,node2-sun:6202"/>
  </topology>
 </notification-server>
 ...
</opmn>
```

Supply the same list for each node in the cluster; each ONS instance will identify itself in the list and ignore that entry.

Note: The `opmn.xml` file must be reloaded for changes to take effect in the OPMN runtime. Run the following command on the affected node to reload `opmn.xml`:

```
opmnctl reload
```

Note that this command will not affect OPMN-managed components, including OHS, OC4J and deployed applications.

Viewing the Status of a Cluster

You can view the current status of the Oracle Application Server components within a cluster, using either `opmnctl` or Application Server Control Console.

- Viewing Cluster Status with opmnctl
- Viewing Cluster Status in Application Server Control Console

Viewing Cluster Status with opmnctl

You can check the status of the cluster using `opmnctl` on any Oracle Application Server node within the cluster.

```
opmnctl @cluster status
```

The output shows the status of the components installed on each active Oracle Application Server instance within the cluster:

```
Processes in Instance: instance1
-------------------+--------------------+---------+---------
ias-component      | process-type       |     pid | status
-------------------+--------------------+---------+---------
OC4J               | home               |   26880 | Alive
HTTP_Server        | HTTP_Server        |   26879 | Alive

Processes in Instance: instance2
-------------------+--------------------+---------+---------
ias-component      | process-type       |     pid | status
-------------------+--------------------+---------+---------
OC4J               | home               |   26094 | Alive
HTTP_Server        | HTTP_Server        |   26093 | Alive
```

Viewing Cluster Status in Application Server Control Console

Click the **Cluster Topology** link in the upper left corner of the Application Server Control Console home page.

The resulting page displays each Oracle Application Server instance that is active within the cluster, as well as the active applications on each instance. Note that you can access an instance or a deployed application within the cluster through this page.

Load Balancing with Oracle HTTP Server

The term *load balancing* refers to the process of distributing incoming service requests over server instances within a cluster. Load balancing in an Oracle Application Server cluster is managed by the `mod_oc4j` module of Oracle HTTP Server (OHS). In this configuration, the OHS instance acts as front-end listener for incoming HTTP/HTTPS requests; `mod_oc4j` then routes each request to an OC4J instance serving the requested application.

In Oracle Application Server Release 3 (10.1.3), load balancing is completely dynamic, and no additional OHS or `mod_oc4j` configuration is required "out of the box". New load-balancing features include:

- Dynamic OC4J instance discovery

  OHS instances are dynamically updated with information on each OC4J instance in the cluster and the applications deployed to it, enabling OHS to route requests to the appropriate instance.

See "Enabling Dynamic Configuration of Application Mount Points" on page 8-26 for details.

- Dynamic routing

  The new release supports a "routing ID" mechanism that allows you to optionally control which OC4J instances that an OHS instance forward requests to, essentially enabling you to control the set of instances which will service requests from specific OHS instances. All OHS and OC4J instances are configured to use a default routing ID upon installation; as such, no configuration is required.

  See "Using Web Server Routing IDs to Control OC4J Request Routing" on page 8-22 for details.

The only requirement is that the ONS servers within the various OHS and OC4J nodes within the cluster be connected using one of the clustering configuration mechanisms outlined in this chapter. See "Configuring a Cluster" on page 8-5 for details.

Using Web Server Routing IDs to Control OC4J Request Routing

Every OHS and OC4J instance in an OPMN-managed installation is assigned a "routing ID" that is passed in at startup from `opmn.xml`. An OHS instance will route incoming Web requests only to OC4J instances that share its routing ID. This means that you can effectively define the set of OC4J instances that a specific OHS instance will route requests to.

A default routing ID is assigned to all component instances, so that upon installation, every OHS instance in a cluster can route requests to any OC4J instance within the cluster.

The routing ID is defined in `opmn.xml` in a `<data>` element where the `id` attribute equals `routing-id`. The `<data>` element entry is a subelement of `<category id="start-parameters">`, which specifies parameters passed to the instance at startup. The default `routing-id` value set for each instance is `"g_rt_id"`.

```
<category id="start-parameters">
  <data id="routing-id" value="g_rt_id"/>
</category>
```

The `<data>` element containing the default routing ID is set within the `<ias-instance>` element, which contains the OPMN configuration data for the Oracle Application Server instance. Because the routing ID is set at this level, the `routing-id` value set in this `<data>` element is applied to all instances of the OHS and OC4J components installed within the OAS instance.

```
<opmn>
 <process-manager>
  ...
  <ias-instance id="instance1" name="instance1">
   ...
    <environment>
     ...
    </environment>
    <module-data>
     <category id="start-parameters">
      <data id="routing-id" value="g_rt_id"/>
     </category>
    </module-data>
   </environment>
   <ias-component id="HTTP_Server">
    ...
```

```
   </ias-component>
   <ias-component id="OC4J">
    ...
   </ias-component>
  </ias-instance>
 </process-manager>
</opmn>
```

However, the routing ID can be set at the individual OHS or OC4J instance level by adding a `<data>` element within the `<category id="start-parameters">` element for the component. This value overrides the routing ID assigned at the Oracle Application Server instance level.

Note that you can specify any string as the value of the `routing-id` attribute; there is no required format for this identifier.

The following entry in `opmn.xml` sets the routing ID for an OHS instance:

```
<opmn>
 <process-manager>
  ...
  <ias-instance id="instance1" name="instance1">
   ...
   <ias-component id="HTTP_Server">
     <environment>
      ...
     </environment>
     <process-type id="HTTP_Server" module-id="OHS">
       <module-data>
        <category id="start-parameters">
          <data id="start-mode" value="ssl-enabled"/>
          <data id="routing-id" value="group_b_id"/>
        </category>
       </module-data>
       <process-set id="HTTP_Server" numprocs="1"/>
      </process-type>
     </ias-component>
  </ias-instance>
 </process-manager>
</opmn>
```

The following entry in `opmn.xml` sets the routing ID for the OC4J `home` instance:

```
<opmn>
 <process-manager>
  ...
  <ias-instance id="instance1" name="instance1">
   ...
    <ias-component id="OC4J">
     <environment>
     </environment>
     <process-type id="home" module-id="OC4J" status="enabled">
       <module-data>
        <category id="start-parameters">
          <data id="java-options" ... />
          <data id="routing-id" value="group_b_id"/>
        </category>
       </module-data>
       <process-set id="HTTP_Server" numprocs="1"/>
       <port id="default-web-site" range="12501-12600" protocol="ajp" />
       <port id="rmi" range="12401-12500"/>
```

```
      <port id="jms" range="12601-12700"/
      <process-set id="default_group" numprocs="1"/>
     </process-type>
    </ias-component>
  </ias-instance>
 </process-manager>
</opmn>
```

Configuring Application Mount Points

To route incoming requests, OHS utilizes a list of application-specific *mount points* that map the URL supplied in a request with the OC4J instance that will service the request. This section includes the following topics on mount point creation:

- Enabling Dynamic Configuration of Application Mount Points
- Changing the Mount Point Configuration Algorithm
- Viewing Mount Point Configuration Data

See the *Oracle HTTP Server Administrator's Guide* for additional details on mount point configuration.

Enabling Dynamic Configuration of Application Mount Points

In previous releases of Oracle Application Server the list of application mount points had to be managed manually in the `mod_oc4j` configuration file, `mod_oc4j.conf`.

In the current release, the mount point list is dynamically updated as new nodes and applications are added to—or removed from—the cluster. This dynamic discovery mechanism is enabled by default and requires no additional configuration.

Using ONS notifications, every OC4J instance within the cluster sends mount point data for each of its deployed applications to `mod_oc4j`, which adds this information to its internal routing table.

The mount point information sent by each OC4J instance to OHS includes:

- The OC4J host address
- OC4J port information, including the Apache JServ Protocol (AJP) listener port

  This value is the lowest available port assigned to AJP in the `opmn.xml` file on the OC4J node.
- The Web module name

  This value is defined as the value of the `name` attribute in the `<web-app>` element defined for the module in the `*-web-site.xml` configuration file the module is bound to.
- The Web context(s) defined for the application

  This value is set in the `root` attribute of the `<web-app>` element defined for the module `*-web-site.xml` configuration file.

> **Note:** Dynamically-configured mount points are not written to the the `mod_oc4j` configuration file (`mod_oc4j.conf`).

When a new application is deployed to an OC4J instance, its mount point information is transmitted to OHS, enabling mod_oc4j to dynamically "discover" the application and begin routing requests to it.

Conversely, when an application is stopped or removed from an OC4J instance, the mod_oc4j routing table is updated to reflect the application's absence, causing mod_oc4j to stop routing requests to the application instance.

Changing the Mount Point Configuration Algorithm

Although dynamic mount point creation is enabled by default, you do have the option of continuing to use manually configured mount points, which is the default mechanism supported in previous releases of Oracle Application Server.

Static mount points are defined in the mod_oc4j configuration file, mod_oc4j.conf, which is installed in the ORACLE_HOME/Apache/Apache/conf directory. By default, OHS will create dynamic mount points as applications are deployed; however, static mount points defined in mod_oc4j.conf will also be honored.

The mount point configuration mechanism to use is specified in the Oc4jRoutingMode parameter in mod_oc4j.conf. Table 8–1 lists the values for this variable. See the *Oracle HTTP Server Administrator's Guide* for details on mount point configuration and using mod_oc4j.conf.

Table 8–1 Oc4jRoutingMode Values

Value	Description
Dynamic	Dynamically configured mount points are used exclusively. Static mount points will be ignored.
Static	Static, manually configured mount points defined in mod_oc4j.conf are used exclusively. Dynamic mount points will not be created for new applications.
DynamicOverride	Both dynamic and static mount points are used. In the event of a conflict, the dynamically configured mount point will be used.
StaticOverride	Both dynamic and static mount points are used; however, in the event of a conflict, the static, manually configured mount point will be used. This is the default mode used, although it is not defined in mod_oc4j.conf by default.

The mod_oc4j.conf example below enables the DynamicOverride mode, in which the dynamic mount points specified will take precedence over static mount points in the event of a conflich:

```
############################################################
# Oracle iAS mod_oc4j configuration file: mod_oc4j.conf #
############################################################

LoadModule oc4j_module libexec/mod_oc4j.so
Oc4jRoutingMode DynamicOverride
<IfModule mod_oc4j.c>
  <Location /oc4j-service>
    SetHandler oc4j-service-handler
  </Location>
    Oc4jMount /j2ee/*
    Oc4jMount /webapp home
    Oc4jMount /webapp/* home
    Oc4jMount /cabo home
```

```
    Oc4jMount /cabo/* home
    Oc4jMount /stressH home
    Oc4jMount /stressH/* home
</IfModule>
```

Viewing Mount Point Configuration Data

You can configure OHS to output mount point configuration data to a Web page generated on the OHS host.

Add the following entry to the OHS configuration file, `httpd.conf`, on the OHS host machine. This file is installed in *ORACLE_HOME*`/Apache/Apache/conf`.

```
<IfModule mod_oc4j.c>
   Oc4jSet StatusUri /oc4j-status
</IfModule>
```

You will now be able to view mount point data by appending the `/oc4j-status` context URI to the OHS server URL:

```
http://ohsHost:ajpPort/oc4j-status
```

For example:

```
http://node1.company.com:7777/oc4j-status
```

The following is sample output displayed in the resulting Web page, with comments:

```
hostname          : node1.company.com
local instance    : node1.company.com
select method     : Round-Robin
select affinity   : None
# OHS routing configuration
routing mode      : Static-Dynamic
routing ID        : g_rt_id

OC4J Dynamic routing
# Applications using dynamic routing

# 'ascontrol' application
application       : ascontrol
  context         : /em
  process (Jgroup): 0

# 'demos' application
application       : demos
  context         : /ojspdemos/jstl, /ojspdemos
  process (Jgroup): 0 (demos)

OC4J Process List

  process,ias instance,host,port,status
  0 : home.node1.company.com, node1.company.com, 12502, ALIVE
    1 : home.node1.company.com, node1.company.com, 12501, ALIVE
    2 : home.node1.company.com, node1.company.com, 12503, ALIVE
```

Replicating Changes Across a Cluster

Because the Distributed Configuration Management (DCM) framework is not provided in Oracle Application Server Release 3 (10.1.3), changes made to individual

configuration files must be manually replicated to each OC4J instance within the cluster. Table 8–2 below summarizes the files that may need to be replicated.

Table 8–2 Configuration Files to Replicate Across a Cluster

File	Location in *ORACLE_HOME*	Data to Replicate/Manage
`application.xml`	`/j2ee/`*`instance`*`/config`	▪ Changes made to configuration data applied by default to all deployed applications. ▪ References to data sources or other shared resources. ▪ Shared library definitions within the `<imported-shared-libraries>` element. Note that the code sources for custom shared libraries must be installed on the OC4J host, and the libraries must be referenced in `server.xml` on the OC4J instance.
`data-sources.xml`	`/j2ee/`*`instance`*`/config`	▪ Configuration data for custom data sources that must be made available to deployed applications.
`default-web-site.xml`	`/j2ee/`*`instance`*`/config`	▪ Secure Web site (HTTPS) configuration, if applicable.
`*-web-site.xml`	`/j2ee/`*`instance`*`/config`	▪ Copy the configuration files for any additional Web sites that will be utilized on the OC4J instance to the specified location. Note that references to Web site configuration files must be added to `opmn.xml` or `server.xml` as outlined in "Creating a New Web Site in OC4J" on page 13-5.
`global-web-application.xml`	`/j2ee/`*`instance`*`/config`	▪ Any new servlet definitions or servlet configuration changes, such as `<init-param>` modifications. ▪ Any modified JSP container properties. See the Oracle Containers for J2EE Support for JavaServer Pages Developer's Guide for details.
`j2ee-logging.xml`	`/j2ee/`*`instance`*`/config`	▪ Any logging configuration changes.
`javacache.xml`	`/j2ee/`*`instance`*`/config`	▪ Any Java cache configuration changes.
`jazn.xml`	`/j2ee/`*`instance`*`/config`	▪ Configuration for either XML- or LDAP-based security providers.
`jazn-data.xml`	`/j2ee/`*`instance`*`/application-deployments/`*`appName`	▪ Replicate the XML-based provider configuration to the specified location for all applications using this provider. Not required for applications using an LDAP-based provider.
`jms.xml`	`/j2ee/`*`instance`*`/config`	▪ Any destination or connection factory additions.
`rmi.xml`	`/j2ee/`*`instance`*`/config`	▪ Any RMI configuration changes, such as logging configuration.

Creating and Managing Additional OC4J Instances

OC4J includes tools for creating or removing additional OC4J instances within an Oracle Application Server instance. Once created, new OC4J instances can be accessed and managed through the Application Server Control Console.

This section includes the following topics:

- Creating an Additional OC4J Instance
- Accessing and Managing a New Instance
- Removing an OC4J Instance

Creating an Additional OC4J Instance

The `createinstance` utility enables you to create additional OC4J instances within an Oracle Application Server instance.

The `createinstance` utility is installed in the `ORACLE_HOME/bin` directory. The syntax is as follows:

```
createinstance -instanceName instanceName [-port httpPort]
```

Note that you must supply an HTTP listener port as the value for `httpPort` when creating a new instance in a standalone OPMN-managed OC4J instance (J2EE Server and Process Management install type.) This HTTP listener port will be set in the `default-web-site.xml` Web site configuration file created for the instance.

As part of the creation process, you will be 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 `home` instance with the `oc4jadmin` user.

Usage Notes:

- The `createinstance` utility can be used regardless of whether the Oracle Application Server instance is in a running or stopped state.
- If OPMN is running, you must reload `opmn.xml` to load the new instance configuration:

  ```
  opmnctl reload
  ```

- If the new OC4J instance will be required to accept ORMI over SSL (ORMIS) requests, you must configure ORMIS in the instance-specific `rmi.xml` file and update `opmn.xml` with the ORMIS port information as described in the *Oracle Containers for J2EE Security Guide*.

Note that you can optionally supply an HTTP port for the value of `-port`. This feature can be used when the Oracle Application Server instance does not include Oracle HTTP Server. Setting an HTTP port makes it possible to access the OC4J instance's "home page" directly.

The new instance will be created within a new *`ORACLE_HOME`*`/j2ee/`*`instanceName`* directory, the same location as the default `home` OC4J instance. A new `<process-type>` element containing the instance configuration will also be added to the `opmn.xml` configuration file.

The following directories and files are generated in the new *ORACLE_HOME*/j2ee/*instanceName* directory structure:

```
applib/
applications/
config/
  contains default versions of all server-level configuration files
config/database-schemas/
  contains all database schema XML files packaged with OC4J
connectors/
  contains RAR files packaged with OC4J
log/
persistence/
```

The new instance does not include the OC4J binary libraries; instead, the instance will utilize the libraries installed in the `home` instance. The `default` application is deployed to the instance; however, binaries and configuration files for other deployed applications, including Application Server Control Console, are not copied to the instance.

Accessing and Managing a New Instance

Once the new instance is started by OPMN, you can access it through the Cluster Topology page in Application Server Control Console.

Log in as the `oc4jadmin` user and supply the password set when the instance was created using the `createinstance` utility.

Once logged in, you can perform the full range of administrator tasks on the instance, including deploying applications to it.

Removing an OC4J Instance

You can delete an OC4J instance by using the `removeinstance` utility, which deletes the directory created for the instance from the `ORACLE_HOME/j2ee/` directory structure and removes configuration data for the instance from `opmn.xml`.

The `removeinstance` utility is installed in the `ORACLE_HOME/bin` directory. The syntax is as follows:

```
removeinstance -instanceName instanceName
```

Usage Notes:

- The OC4J instance to be deleted must be in a stopped state.
- If OPMN is running when the tool is in use, you must invoke `opmnctl reload` to reload the updated `opmn.xml` into the runtime.
- The default `home` instance cannot be deleted.

9

Application Clustering in OC4J

This chapter discusses the clustering framework provided in OC4J 10*g* Release 3 (10.1.3). It includes the following topics:

- What Is Clustering in OC4J?
- Configuring Application Clustering

What Is Clustering in OC4J?

OC4J provides a flexible framework for creating a clustered environment for development and production purposes. In this context, a *cluster* is defined as two or more OC4J server nodes hosting the same set of applications. The OC4J clustering framework supports:

- Replication of objects and values contained in an HTTP session or a stateful session Enterprise JavaBean (SFSB) instance.
- In-memory replication using multicast or peer-to-peer communication, or persistence of state data to a database.
- Load balancing of incoming requests across OC4J instances.
- Transparent failover across applications within the cluster.
- Configuration within an OC4J instance at either the global server or application level.

A new `<cluster>` element, which contains a number of new subelements, has been added to the XML schema definition for these files to provide a single mechanism for clustering management. See "Overview of the <cluster> Element" on page 9-10 for descriptions of this element and its subelements.

How Does Clustering Differ from Previous OC4J Releases?

The following are no longer included in the clustering framework in OC4J 10*g* (10.1.3).

"Islands" No Longer Supported

The notion of *islands*, part of the clustering framework in previous OC4J releases, is no longer supported in OC4J.

In previous releases, an island was essentially a group of OC4J instances within a cluster across which HTTP session data was replicated. Although islands reduced overhead by not replicating data across the entire cluster, they increased configuration and management overhead. In addition, islands were only applicable to Web applications; EJB applications could not utilize the island configuration.

In OC4J 10*g* (10.1.3), you can still effectively limit the number of instances to replicate data by using the `write-quota` attribute of the `<cluster>` element. This attribute makes it possible to control the scope of state replication.

See "Overview of the <cluster> Element" on page 9-10 for details on the `write-quota` attribute.

loadbalancer.jar No Longer Used

The `loadbalancer.jar`, which provided load balancing functionality in previous OC4J releases, was deprecated in the previous release of OC4J and has been removed from the current release.

Deprecated Clustering-Specific XML Elements

The following XML elements are deprecated in OC4J 10*g* (10.1.3) and should no longer be used to configure clustering:

- The `<cluster-config>` element in `server.xml`, the OC4J configuration file
- The `cluster-island` attribute of the `<web-site>` element in a `*-web-site.xml` Web site configuration file

The new `<cluster>` element is now used for all cluster management.

Configuring Application Clustering

Clustering is enabled by adding the `<cluster>` element to the `orion-application.xml` file of each application to be clustered within an OC4J instance. For deployed applications, this file is located in the *`ORACLE_HOME`*`/j2ee/`*`instance`*`/application-deployments/`*`applicationName`* directory. See "Overview of the <cluster> Element" on page 9-10 for descriptions of this element and its subelements.

This section includes the following topics:

- Enabling Clustering
- Setting Replication Policies
- Managing the Number of Nodes to Replicate To
- Synchronous versus Asynchnronous Replication
- Configuring Multicast Replication
- Configuring Peer-to-Peer Replication
- Configuring Database Replication
- Disabling Clustering
- Overview of the <cluster> Element

Enabling Clustering

Clustering can be enabled globally for all applications running within an OC4J instance, as well as on a per-application basis.

- **Enabling clustering for all applications**

  Clustering can be enabled by default for all applications deployed to the OC4J instance through *`ORACLE_HOME`*`/j2ee/home/config/application.xml`, the configuration file for the global `default` application. All other applications

deployed into the OC4J instance inherit default properties from this application, including the clustering configuration.

- **Enabling clustering for a specific application**

  Clustering is defined in the application-specific *ORACLE_HOME*/j2ee/home/application-deployments/*app_name*/orion-application.xml file. Settings in this file override the global configuration, as well as the configuration inherited from a parent application.

  Note: Clustering can also be configured at the time the application is deployed by using Oracle Enterprise Manager 10*g* Application Server Control Console, through either the deployment tasks or the deployment plan editor.

  See the *Oracle Containers for J2EE Deployment Guide* for details.

Managing a clustered environment is an entirely manual process, as no distributed management tools are provided in the current release of Oracle Application Server. This means that any changes made to a particular application's orion-application.xml file on one OC4J server must be manually replicated to the corresponding XML file on all server nodes within the cluster.

At the application level, clustering can be configured at the time the application is deployed into an OC4J instance using the deployment plan editor, which sets values in the application's orion-application.xml file. See the *Oracle Containers for J2EE Deployment Guide* for details on using the deployment plan editor. Note that the deployment plan editor does not support configuration of the global default application, and configuration at the global level is completely manual.

Important: An empty <distributable /> tag must be added to the web.xml file for all Web modules that are part of an application configured to use clustering.

Post-deployment, this J2EE standard Web module descriptor is found in the *ORACLE_HOME*/j2ee/home/applications/*app_name*/*web_module*/WEB-INF directory within OC4J.

Setting Replication Policies

A replication policy defines when replication of HttpSession or a stateful session bean state occurs, and whether all attributes and variable values or only changed values are replicated. Replication can be an expensive process, and replicating data too frequently can affect server performance. On the other hand, replicating data too infrequently can result in lost data in the event of server failure.

The replication policy applied to all Web modules and EJB components within an application is specified in the <replication-policy> element within the application's orion-application.xml configuration file. The syntax of this element is as follows:

```
<replication-policy trigger="onSetAttribute|onRequestEnd|onShutdown"
 scope="modifiedAttributes|allAttributes" />
```

- The trigger attribute specifies when replication occurs. By default, the onRequestEnd policy is applied, as it provides frequent replication of data while ensuring that data is not lost if the JVM terminates unexpectedly.

See Table 9–1 for an overview of `trigger` attribute values.

- The `scope` attribute defines what data is replicated: Either all attribute or variable values, or only changed values. By default, only modified HTTP session attributes are replicated; for stateful session beans, all member variables are replicated.

  See Table 9–2 for an overview of `scope` attribute values.

Table 9–1 <replication-policy> trigger Attribute Values

trigger Value	HttpSession	Stateful Session Bean
`onSetAttribute`	Replicate each change made to an HTTP session attribute at the time the value is modified. From a programmatic standpoint, replication occurs each time `setAttribute()` is called on the `HttpSession` object. This option can be resource intensive in cases where the session is being extensively modified.	Not applicable.
`onRequestEnd` (default)	Queue all changes made to HTTP session attributes, then replicate all changes just before the HTTP response is sent.	Replicate the current state of the bean after each EJB method call. The state is replicated frequently, but offers higher reliance.
`onShutdown`	Replicate the current state of the HTTP session whenever the JVM is terminated gracefully, such as with Ctrl-C. State is not replicated if the host is terminated unexpectedly, as in the case of a system crash. Because session state was not previously replicated, all session data is sent across the network at once upon JVM termination, which can impact network performance. This option can also significantly increase the amount of time needed for the JVM to shut down.	Replicate the current state of the bean whenever the JVM is terminated gracefully. State is not replicated if the host is terminated unexpectedly, as in the case of a system crash. Because bean state was not previously replicated, all state data is sent across the network at once upon JVM termination, which can impact network performance. This option may also significantly increase the amount of time needed for the JVM to shut down.

Table 9–2 <replication-policy> scope Attribute Values

scope Value	HttpSession	Stateful Session Bean
`modifedAttributes` (default)	Replicate only modified HTTP session attributes; that is, values changed by calling `setAttribute()` on the `HttpSession` object.	Not applicable.
`allAttributes`	Replicate all attribute values set on the HTTP session.	Replicate all member variable values set on the stateful session bean.

Note that the <replication-policy> element in orion-application.xml does not allow you to distinguish between Web and EJB components within an application. However, you can specify a different replication policy for an EJB component in the replication attribute of the <session-deployment> element within the component-specific orion-ejb-jar.xml configuration file.

See Table 9–3 for valid values for the replication attribute. For example:

```
<session-deployment name="MyStatefulVM" replication="onShutdown" />
<session-deployment name="MyEntity2" replication="onRequestEnd" />
```

The values in this file overrides the corresponding settings in orion-application.xml, effectively allowing you to set the replication policy for an EJB component in orion-ejb-jar.xml and the policy for Web components in orion-application.xml.

Table 9–3 Stateful Session EJB Replication Policy Configuration

replication Value	Description
onRequestEnd (default)	Replicate the current state of the bean after each EJB method call. The state is replicated more frequently, but offers higher reliability in the event of host failure. This is the default value.
onShutdown	Replicate the current state of the bean whenever the JVM is terminated gracefully. State is not replicated if the host is terminated unexpectedly, as in the case of a system crash or a "kill -9" invocation in Unix/Linux.
none	Do not replicate data.

Managing the Number of Nodes to Replicate To

You can effectively limit the number of group members to which state data is replicated by using the write-quota attribute of the <cluster> element. This functionality makes it possible to reduce network traffic and related overhead by controlling the scope of state replication.

The default value for write-quota is 1, indicating that state will be replicated to one other OC4J node within the cluster. Note that this value is valid only in a cluster with just two nodes; for clusters with three or more nodes, set this value to 2 or higher. This configuration ensures that in the event that one node goes down, state will be replicated to at least two other nodes.

To replicate state to all member nodes within the cluster, you must specify the total number of nodes within the cluster as the value of write-quota.

Synchronous versus Asynchnronous Replication

By default, OC4J nodes will replicate data to other nodes asynchronously. However, you can enable synchronous replication by including the <synchronous-replication> subelement within the <cluster> element. This will force a replicating OC4J node to wait for an acknowledgement that the data was received from at least one other peer node before continuing with replication.

Configuring Multicast Replication

Multicast IP replication is the default replication protocol used in a standalone OC4J installation. In this mode, OC4J uses multicast packages to send and receive HTTP session and stateful session bean state changes. These packages are sent over the network to be picked up by other OC4J processes using the same multicast address

and port. Lost messages are identified and retransmitted, providing a reliable transmission service.

The configuration must specify the same multicast address and port on all OC4J instances. The default values used by OC4J multicast are `230.230.0.1` for the address and `45566` for the port. These values can be changed in the appropriate XML configuration file, if necessary.

Multicast replication can be enabled between multiple application instances simply by adding an empty `<cluster>` element to `orion-application.xml` file for each instance:

```
<orion-application ...>
  ...
  <cluster/>
</orion-application>
```

The next example specifies a new multicast address and port, using the `ip` and `port` attributes.

Note the optional `bind_addr` attribute, which can be used to specify which Network Interface Card (NIC) to bind to. This is useful if you have OC4J host machines with multiple network cards, each with a specific IP address, and you wish to define which NIC is used to send and receive the multicast messages.

```
<orion-application ...>
  ...
  <cluster allow-colocation="false">
    <replication-policy trigger="onShutdown" scope="allAttributes" />
    <protocol>
      <multicast ip="225.130.0.0" port="45577" bind_addr="226.83.24.10" />
    </protocol>
   </cluster>
</orion-application>
```

Using an Existing JavaGroups Configuration for Multicast Replication

The multicast-based and peer-to-peer-based replication mechanisms provided by OC4J are built on the JavaGroups communication protocol stack. Ideally, you should use one of these OC4J mechanisms to provide in-memory replication of state data, as they utilize OC4J-specific configurations.

However, you do have the option of utilizing your own JavaGroups configuration within the OC4J clustering framework. This feature is enabled by specifying one of the following in the `<property-config>` subelement within the `<cluster>` element:

- A string containing the JavaGroups configuration properties
- A URL to an XML configuration file containing this information

See the description of `<javagroups-config>` in "Overview of the <cluster> Element" on page 9-10 for details.

Configuring Peer-to-Peer Replication

OC4J supports replication in a peer-to-peer (P2P) topology, using TCP to establish connections between nodes within the cluster. The state data held in each application instance is then unicast to each node.

Two peer-to-peer configurations are supported:

- Dynamic peer-to-peer, in which Oracle Process Manager and Notification Server (OPMN) is used to enable peer nodes to dynamically discover and communicate with one another. This configuration is the default used in an Oracle Application Server environment where OPMN is used to manage the various components, including OC4J.

  See "Configuring Dynamic OPMN-Managed Peer-to-Peer Replication" for details.

- Static peer-to-peer, in which each node in the cluster is explicitly configured to recognize at least one other peer node. This configuration is supported only in a standalone OC4J environment, with a relatively small number of standalone OC4J instances clustered together.

  See "Configuring Static Peer-to-Peer Replication" for details.

Configuring Dynamic OPMN-Managed Peer-to-Peer Replication

In an Oracle Application Server environment, Oracle Process Manager and Notification Server (OPMN) is utilized to provide "dynamic" peer-to-peer replication. In this replication model, each OC4J node registers itself with OPMN. The node then queries OPMN for the list of available nodes, enabling it to dynamically discover and communicate with other nodes within the cluster.

Note: To use this feature, all OC4J nodes hosting the application must be first be members of a cluster utilizing either the OPMN dynamic multicast discovery or static discovery server mechanisms.

See "Supported Clustering Models" on page 8-2 for details.

Each OC4J process sends periodic ONS (heartbeat) messages to OPMN to inform OPMN of current status, enabling OPMN to maintain a real-time list of available peer nodes, and to notify nodes when one has failed. In the event that a node is lost, another node is able to service its requests.

```
<orion-application ...>
  ...
  <cluster>
    <protocol>
      <peer>
       <opmn-discovery />
      </peer>
    </protocol>
  </cluster>
</orion-application>
```

Configuring Static Peer-to-Peer Replication

In this configuration, the host address and port of at least one other peer node are supplied to enable for peer-to-peer communication. As a node becomes aware of each of its peers, it also becomes aware each peer's peer(s) - with the end result that all of the nodes in the cluster become aware of one another.

The key challenge in this configuration is in ensuring that host and port definitions are kept up to date, which may present a significant management effort. The following elements and attributes affect the configuration:

- The `start-port` attribute of the `<peer>` element specifies the initial port on the host that the local OC4J process will try to bind to for peer communication. If this

port is not available, OC4J will continue to increment this port until an available port is found.

- The <node> element specifies a peer node. The host and port attributes of the element define the name of the node address and the port that will be used for peer communication.
- The range attribute of the <peer> element applies to the ports specified in each <node> element - not to the value of the start-port attribute. The range attribute defines the number of times to increment the port value if the specified port is not available on a node.

The following example illustrates static peer-to-peer configurations as specified in the orion-application.xml application deployment descriptor deployed with the sample application to three cluster nodes.

In this configuration, each node specifies one other node as its peer. The result is that all of the nodes within the cluster are able to establish connections with one another. Note that this scenario will work only if each node is started in succession; that is, www1.company.com must be started before www2.company.com. Otherwise, www2.company.com will not be able to "see" www1.company.com.

1. First, www1.company.com specifies www2.company.com as its peer:

```
<orion-application ...>
  ...
  <cluster>
    <protocol>
      <peer start-port="7900" range="10" timeout="6000">
        <node host="www2.company.com" port="7900" />
      </peer>
    </protocol>
  </cluster>
</orion-application>
```

2. Next, www2.company.com specifies www3.company.com as its peer:

```
<orion-application ...>
  ...
  <cluster>
    <protocol>
      <peer start-port="7900" range="10" timeout="6000">
        <node host="www3.company.com" port="7900" />
      </peer>
    </protocol>
  </cluster>
</orion-application>
```

3. Finally, www3.company.com specifies www1.company.com as its peer:

```
<orion-application ...>
  ...
  <cluster>
    <protocol>
      <peer start-port="7900" range="10" timeout="6000">
        <node host="www1.company.com" port="7900" />
      </peer>
    </protocol>
  </cluster>
</orion-application>
```

An alternative configuration could have all of the nodes specifying the same node as a peer. For example, you could have the `www1.company.com` and `www3.company.com` nodes both specify `www3.company.com` as a peer. In this configuration, `www2.company.com` would have to be the first node started; the other nodes would then connect to this node, and establish connections with one another.

Configuring Database Replication

The new clustering framework provides the ability to replicate an HTTP session and stateful session bean state to a database. Data is persisted outside of the clustered OC4J framework, enabling the entire session to be recovered in the event of a catastrophic failure of all of the OC4J instances within the cluster. Note that the full HTTP session or stateful session bean object is replicated to the database.

The connection to the database is created using a *data source*, which is specified in the `data-source` attribute of the `<database>` subelement of `<protocol>`. Set the value of the `data-source` attribute to the data source's `jndi-name` as specified in `data-sources.xml`.

The data source specified must already exist within the OC4J instance. See the *Oracle Containers for J2EE Services Guide* for details on creating and using data sources.

The following example configures the application to replicate data to the database accessed through the "MyOracleDS" data source.

```
<orion-application ...>
  ...
  <cluster>
    <protocol>
      <database data-source="jdbc/MyOracleDS"/>
    </protocol>
  </cluster>
</orion-application>
```

Session data is persisted to the following tables in the database:

- `OC4J_HTTP_SESSION`, which stores metadata for an HTTP session
- `OC4J_HTTP_SESSION_VALUE`, which stores the values set by the application user on the HTTP session
- `OC4J_EJB_SESSION`, which stores the current state of a stateful session bean

The tables are created by OC4J the first time database replication is invoked. See Appendix C, "Overview of the Session State Tables" for details on the table schema.

The length of time session data is stored in the database is based on the session's time-to-live (TTL) value. A session is considered expired when the difference between the current database time and the time the session was last accessed is greater than the session timeout value. The actual equation for determining a session's TTL is:

```
(Current Database Time - Last Accessed Time) > Max Inactive Time
```

Expired sessions are removed from the database on the next execution of the OC4J task manager. See "Configuring the OC4J Task Manager" on page 11-1 for instructions on setting the task manager interval.

In the event that the OC4J server terminates without proper session termination, orphan records will be created in the database. These records will also be deleted the next time the task manager runs.

Disabling Clustering

Clustering can be disabled globally or for a specific application using the Boolean `enabled` attribute of the `<cluster>` element. Note that setting this attribute to `false` in an application's `orion-application.xml` file effectively removes the application from the cluster.

Overview of the <cluster> Element

The `<cluster>` element serves as the single mechanism for clustering configuration. It is used exclusively in the `ORACLE_HOME/j2ee/home/config/application.xml` file to configure clustering at the global level, and in application-specific `orion-application.xml` files for application-level clustering configuration.

<cluster>

Contains the clustering configuration for an enterprise application running within an OC4J instance.

Subelements of `<cluster>`:

```
<property-config>
<flow-control-policy>
<replication-policy>
<protocol>
<synchronous-replication>
```

Attributes:

- `enabled`: Whether clustering is enabled for the application. The default is `true`. Note that setting this value at the application level overrides the value inherited from the parent application, including the `default` application.
- `group-name`: The name to use when establishing the replication group channels. If not supplied, the application name as defined in `server.xml`, the OC4J server configuration file, is used by default, and new group channels are created for each enterprise application.

  If a value is specified, the application and all child applications will use the channels associated with this group name.

  Note that this attribute is ignored if the `<database>` tag is included.
- `allow-colocation`: Whether to allow application state to be replicated to a node residing on the same host machine.

  The default is `true`. However, this attribute should be set to `false` if multiple hosts are available.

  If multiple OC4J instances are instantiated on the same machine, different listener ports must be specified for each instance in the `default-web-site.xml`, `jms.xml` and `rmi.xml` configuration files.
- `write-quota`: The number of other group members the application state should be replicated to. This attribute makes it possible to reduce overhead by limiting the number of nodes state is written to, similar to the "islands" concept used in previous OC4J releases.

  The default is `1` node.

  Note that this attribute is ignored if the `<database>` tag is included.

- `cache-miss-delay`: The length of time, in milliseconds, to wait in-process for another group member to respond with a session if the session cannot be found locally. If the session cannot be found, the request will pause for the entire length of time specified.

  The default is `1000` milliseconds. In installations where heavy request loads are expected, this value should be increase, for example to `5000`. Setting this value higher also prevents the OC4J instance from creating a replica of session data within itself if `allow-colocation` is set to `true`.

  Note that this attribute is ignored if the `<database>` tag is included.

<property-config>

Contains data required to use the JavaGroups group communication protocol to replicate session state across nodes in the cluster.

Attributes:

- `url`: A link to a JavaGroups XML configuration file.
- `property-string`: A string containing the properties that define how the JavaGroups JChannel should be created.

<replication-policy>

The replication policy to apply, which defines when replication of data occurs and what data is replicated.

Attributes:

- `trigger`: The frequency at which replication occurs. See Table 9–1 on page 9-4 for the values for this attribute.
- `scope`: What data is replicated. See Table 9–2 on page 9-4 for the values for this attribute.

<protocol>

Defines the mechanism to use for data replication. Note that only one can be specified.

Subelements:

```
<multicast>
<peer>
<database>
```

<multicast>

Contains the configuration required to use multicast communication for replication. This is the default protocol used.

Attributes:

- `ip`: The multicast address to use. The OC4J default is `230.230.0.1`.
- `port`: The multicast port to use. The OC4J default is port `45566`.
- `bind_addr`: The Network Interface Card (NIC) to bind to. This is useful if you have OC4J host machines with multiple network cards, each with a specific IP address.

<peer>

Contains the configuration required to use peer-to-peer (P2P) communication for replication.

Subelements:

```
<opmn-discovery>
<node>
```

Attributes:

- `start-port`: The initial port on the node to attempt to allocate for peer communication. OC4J will continue to increment this value until an available port is found. The default is port `7800`. Valid only for configuring static peer-to-peer replication in a standalone OC4J installation.
- `range`: The number of times to increment the port value specified in each <node> subelement while looking for a potential peer node. The default is `5` increments. Valid only for configuring static peer-to-peer replication in a standalone OC4J installation.
- `timeout`: The length of time, in milliseconds, to wait for a response from a peer while looking for a potential peer node. The default is `3000` milliseconds. Valid only for configuring static peer-to-peer replication in a standalone OC4J installation.
- `bind_addr`: The Network Interface Card (NIC) to bind to. This is useful if you have OC4J host machines with multiple network cards, each with a specific IP address.

<opmn-discovery>

Configures OC4J to use "dynamic" peer-to-peer replication in an Oracle Application Server environment.

<node>

Contains the host name and port of a node to poll if using static peer-to-peer communication. One or more instances of this element can be supplied within a `<peer>` element.

Attributes:

- `host`: The host name of the peer node as a URL.
- `port`: The port on the node to use for peer-to-peer communication. The default is port `7800`.

<database>

Contains the connection information required to persist state data to a database.

Attributes:

- `data-source`: The name of a data source containing the database connection information. This must be the value of the data source's `jndi-name` as specified in `data-sources.xml`.

<flow-control-policy>

Controls the amount of memory to allocate to the handling of clustering messages during replication. This element is intended to prevent out-of-memory errors by gating the amount of data (bytes) sent from one OC4J node to another during replication.

Attributes:

- `enabled`: Whether flow control is enabled. The default is `true`.

- `max-bytes`: The maximum number of bytes the receiving OC4J node can accept. After this value is reached, the sending OC4J node must wait for an acknowledgement from the receiver before additional messages can be received. The default value is `500000`.
- `min-bytes`: The minimum number of bytes the receiving OC4J node can accept without triggering an acknowledgement that more bytes should be sent. If the bytes received is below this value, the receiver will acknowledge that it can accept more bytes from the sender. The default is `0`.
- `threshold`: If `min-bytes` is not specified, this factor value is applied to incoming requests to determine the value of that attribute. The default value is `.25`.

<synchronous-replication>

If included, an OC4J node replicating application data will wait for an acknowledgement that the data update was received from at least one other peer node before continuing with replication. This element is optional; the default behavior is for nodes to continue replicating data to other nodes asynchronously.

Attributes:

- `timeout`: The length of time, in milliseconds, to wait for a response from a peer node. If this value is exceeded, replication should continue, although no acknowledgement will be sent. The default value is 10000 milliseconds (10 seconds).

10

Logging in OC4J

This chapter provides instructions on using the system and application logging features available in OC4J. It covers the following:

- Log Files Generated by OC4J
- Using Plain Text File Logging
- Using Oracle Diagnostic Logging (ODL)

Log Files Generated by OC4J

Each OC4J process generates a number of log files to aid in troubleshooting. If there are multiple processes running for an OC4J instance, multiple sets of log files are generated.

There are two types of log files that can be generated by OC4J:

- **Plain text log files**

  Plain text logs are the default log files used for OC4J components, and are ideal for use in a development environment. The messages logged in these text files can be read with any editor, including the Oracle Enterprise Manager 10*g* Application Server Control Console.

- **Oracle Diagnostic Logging (ODL) Log Files**

  The messages logged in these files use an XML format that is viewable through Application Server Control Console. The key benefit of ODL is that it supports log file rotation.

Log files are generated in different locations, depending on the component or application that data is being recorded for. The logging configuration for each component or application is defined in component-specific XML configuration files.

Table 10–1 lists the names and locations of the various log files generated, as well as the XML configuration file containing the logging configuration for each component. Unless otherwise indicated, all paths indicated are within *ORACLE_HOME*/j2ee/home for standalone OC4J, or *ORACLE_HOME*/j2ee/*instance_name* for OPMN-managed instances.

Table 10–1 List of Log Files Generated for OC4J

Component	Configuration File	Default Log File Name and Location

OC4J components using Java logging	See "Configuring OC4J Logging" on page 10-6 for details on configuring this ODL-formatted log file.	Standalone OC4J:
		`/log/oc4j/log.xml`
		
		OPMN-managed OC4J:
		`/log/`*`instance`*`_default_group_1/oc4j/log.xml`

Application Server Control Console	`/application-deployments/ascontrol/orion-application.xml`	Standalone OC4J:
		`/log/ascontrol-application.log`
		
		OPMN-managed OC4J:
		`/log/`*`instance`*`_default_group_1/ascontrol-application.log`

Applications deployed into OC4J	`/application-deployments/`*`app_name`*`/orion-application.xml`	Standalone OC4J:
		`/application-deployments/`*`app_name`*`/application.log`
		
		OPMN-managed OC4J:
		`/application-deployments/`*`app_name`*`/`*`instance`*`_default_group_1/application.log`

Global (`default`) application	`/config/application.xml`	Standalone OC4J:
		`/log/global-application.log`
		
		OPMN-managed OC4J:
		`/log/`*`instance`*`_default_group_1/global-application.log`

Default Web site access logging	`/config/default-web-site.xml`	Standalone OC4J:
		`/log/default-web-access.log`
		
		OPMN-managed OC4J:
		`/log/`*`instance`*`_default_group_1/default-web-access.log`

OC4J server	`/config/server.xml`	Standalone OC4J:
		`/log/server.log`
		
		OPMN-managed OC4J:
		`/log/`*`instance`*`_default_group_1/server.log`

Table 10–1 (Cont.) List of Log Files Generated for OC4J

Component	Configuration File	Default Log File Name and Location
JMS	/config/jms.xml	Standalone OC4J: /log/jms.log OPMN-managed OC4J: /log/*instance*_default_group_1/jms.log
RMI	/config/rmi.xml	Standalone OC4J: /log/rmi.log OPMN-managed OC4J: /log/*instance*_default_group_1/rmi.log
OPMN	ORACLE_HOME/opmn/conf/opmn.xml	ORACLE_HOME/opmn/logs

Using Plain Text File Logging

Plain text logging is the default format used in OC4J.

This mechanism separates messages in alignment with the XML files. However, instead of writing to multiple log files of the same size, all messages for that component are written into a single log file. The following topics describe how to use text logging:

- Enabling/Disabling Text File Logging
- Managing Text Log Files
- Viewing Text Log Files

Enabling/Disabling Text File Logging

Text logging is enabled or disabled through elements in the XML configuration files listed in Table 10–1, except for the `default-web-site.xml` file. (See "Configuring Web Site Access Logging" on page 13-12 for details on configuring Web site access logging.)

Logging is enabled via the `<file>` subelement of the `<log>` element of the XML configuration file for each component. The element contains a single `path` attribute which specifies the name and optionally the location of the log file generated:

```
<log>
  <file path="application.log" />
</log>
```

To turn off text logging for a component, remove or comment out the `<file>` element from the appropriate configuration file. If you do not remove this line and you enable ODL, both logging options will be enabled.

For example, to disable text logging for an application, comment out the following element in the application's `orion-application.xml` file:

```
<!--
<log>
  <file path="application.log" />
</log>
-->
```

Note that although both ODL and text logging can be enabled simultaneously, one of these options should be disabled to save disk space.

Managing Text Log Files

It is important to monitor your log files, as text logging does not have any imposed size limits or log rotation capability. If left unchecked, log files will continue to grow and can overrun the disk.

The only way to manage these files is to stop OC4J, remove the files, and then restart OC4J to start the log files over.

Viewing Text Log Files

All text log files are generated by default in the locations listed in Table 10–1, " List of Log Files Generated for OC4J" on page 10-2. Text log files are identified by the `log` extension.

Note that text log files generated for OC4J components can be viewed through Application Server Control Console, as follows:

1. Click the **Logs** link at the bottom of any Application Server Control Console page.
2. Expand **OC4J**.
3. Expand **<instanceName>**. The default instance name is `home`.

Note that text log files for deployed J2EE applications cannot be viewed through Application Server Control Console.

Using Oracle Diagnostic Logging (ODL)

The *Oracle Diagnostic Logging* framework, or *ODL*, provides plug-in components that complement the standard Java framework to automatically integrate log data with Oracle log analysis tools.

In the ODL framework, log files are formatted as XML documents, enabling logs to be parsed and reused by other Oracle Application Server and custom-developed components, including Application Server Control Console. Another key benefit of ODL is that unlike in text-based logging, log file rotation is supported.

- Enabling/Disabling ODL
- Managing ODL Log Files
- Viewing ODL Log Files

Enabling/Disabling ODL

ODL is enabled by adding the `<odl>` element within the `<log>` element in any of the XML files listed in Table 10–1.

Notes:

- You can enable ODL for an application at the time the application is deployed by setting values for `odls` in the `log` property through the deployment plan editor.

  See the *Oracle Containers for J2EE Deployment Guide* for details on configuring an application using the deployment plan editor.
- ODL for Web sites uses a different configuration. See "Configuring Web Site Access Logging" on page 13-12 for details on configuring Web site access logging.
- Both ODL and text file logging can be enabled simultaneously. However, you should disable one of these options to save disk space.

The `<odl>` element has the following attributes. All are required.

- `path`: The path to the directory where the `log.xml` files for this component will be generated.

  Important:

  Specify the path as `../log/`*`appName`*, as shown in the example below. This path is required to enable log files to be viewed through Application Server Control Console.

- `max-file-size`: The maximum size, in kilobytes, that an individual log file is allowed to grow to. When this limit is reached, a new log file is generated.
- `max-directory-size`: Sets the maximum size, in kilobytes, allowed for the log file directory. When this limit is exceeded, log files are purged, beginning with the oldest files.

For example, the following entry in the `petstore` application's `orion-application.xml` file will cause `log.xml` files to be generated for this application. It will also set log files to a maximum of 1000 KB and the directory maximum to 10,000 KB.

```
<log>
  <odl path="../log/petstore/" max-file-size="1000" max-directory-size="10000" />
</log>
```

Using this configuration, `petstore` log files will be generated in the following locations, depending on your OC4J installation.

- Standalone OC4J:

  Log files will be generated in `ORACLE_HOME/j2ee/home/application-deployments/log/petstore`.
- OPMN-managed OC4J:

  Files will be generated in an OC4J instance- specific directory named *`ORACLE_HOME`*`/j2ee/`*`instance_name`*`/application-deployments/log/`*`instance_name`*`_default_group_1/petstore`.

Managing ODL Log Files

A key benefit of the ODL framework is that it provides support for managing log files, including log file rotation. The maximum log file size and the maximum size of log directories can also be defined.

When you enable ODL, each new message goes into the current log file, named `log.xml`. When the log file is full—that is, the log file size maximum is reached—then it is copied to an archival log file, named `log`*N*`.xml`, where N is a number starting at one. When the last log file is full, the following occurs:

1. The oldest log file is erased to provide space in the directory.
2. The `log.xml` file is written to the latest `log`*N*`.xml` file, where N increments by one over the most recent log file.

Viewing ODL Log Files

ODL-formatted log files can be viewed by clicking the **Logs** link in the Web-based Application Server Control Console, allowing administrators to aggregate and view the logging output generated by all components and applications running within OC4J from one centralized location.

ODL log files are identified in the Log Files page by the `.xml` extension.

1. Click the **Logs** link at the bottom of any Application Server Control Console page.
2. Expand **OC4J**.
3. Expand **<instanceName>**. In both standalone OC4J and OAS, the default instance name is `home`.

- To view the OC4J log files, expand **Diagnostic Message Logs**, then open `log.xml`.
- To view ODL logs for a specific J2EE application:
  - Expand the **Application <applicationName>** node.
  - Expand **Diagnostic Message Logs**. Open and view the `log.xml` file generated within this director.

Configuring OC4J Logging

The various components of OC4J utilize Java loggers that write to the OC4J log file. The OC4J log file is generated in XML format using the Oracle Diagnostic Logging framework and can be viewed through Application Server Control Console.

The section covers the following topics:

- Using and Configuring the OC4J Component Loggers
- Viewing the OC4J Log File
- Configuring the oracle Logger

Using and Configuring the OC4J Component Loggers

OC4J provides a number of component loggers that write to the OC4J log file (`log.xml`). The available component loggers can be viewed and configured through the Logger Configuration page in Application Server Control Console.

The Java log level can be set for each individual component logger. If set to NULL, a logger inherits the log level set for its parent.

Therefore, the default level for all loggers is INFO—which maps to the NOTIFICATION Java log level—as that is the default value inherited from the oracle logger. See "Configuring the oracle Logger" on page 10-8 for details on changing this default value.

Note that the log level set on a logger through the Logger Configuration page is not persisted, but is applied to the OC4J runtime only; when OC4J is restarted, the log level reverts back to the default setting inherited from the parent logger.

Table 10–2 below illustrates the log levels that can be set through Application Server Control Console and the ODL *message type:log level* that each maps to.

Table 10–2 OC4J Logger Log Levels

Java Log Level	ODL Message Type:Log Level	ODL Description
NULL		The logger will inherit the log level set for its parent.
SEVERE	ERROR:1	Log system errors requiring attention from the system administrator.
WARNING	WARNING:1	Log actions or conditions discovered that should be reviewed and may require action before an error occurs.
INFO	NOTIFICATION:1	Log normal actions or events. This could be a user operation, such as "login completed" or an automatic operation such as a log file rotation.
CONFIG	NOTIFICATION:16	Log configuration-related messages or problems.
FINE	TRACE:1	Log trace or debug messages used for debugging or performance monitoring. Typically contains detailed event data.
FINER	TRACE:16	Log fairly detailed trace or debug messages.
FINEST	TRACE:32	Log highly detailed trace or debug messages.

To configure OC4J component loggers through Application Server Control Console:

1. Click the **Administration** link.
2. Click **Logger Configuration**.
3. Set Log Level to a value listed in the left-hand column of Table 10–2 above.
4. Click **Apply** to apply your changes to the OC4J runtime.

Viewing the OC4J Log File

The OC4J log file can be viewed through Application Server Control Console. To view the file:

1. Click the **Logs** link at the bottom of any Application Server Control Console page.
2. Expand **OC4J**.
3. Expand **<instanceName>**. In both standalone OC4J and OAS, the default instance name is `home`.
4. Expand **Diagnostic Message Logs**.

As with all ODL log files, each new message goes into the current log file, named `log.xml`. Once the maximum size is reached, the log is copied to an archival log file, named `log`*N*`.xml`, where N is a number starting at 1.

Configuring the oracle Logger

The configuration for the `oracle` logger is defined in `j2ee-logging.xml`, which is installed in the *`ORACLE_HOME`*`/j2ee/`*`instance`*`/config` directory.

In the current release, the `j2ee-logging.xml` configuration file must be edited by hand. Restart OC4J after making any changes to this file.

The configuration file contains two elements within the `<logging-configuration>` root element:

- `<log_handlers>`

  This element includes a `<log_handler>` element defining `oc4j-handler`. The following properties are specified in a `<property>` subelement:

  - `path`: Specifies the directory in which the Handler will generate log files. Do not modify this value.
  - `maxFileSize`: Sets the maximum size, in bytes, that any log file in the directory will be allowed to grow to. When a file exceeds this limit, a new file is generated.
  - `maxLogSize`: Sets the maximum size, in bytes, allowed for the log file directory. When this limit is exceeded, log files are purged, beginning with the oldest files.

- `<loggers>`

  This element includes a `<logger>` element defining the following:

  - `name`: The Logger name. Do not modify this value.
  - `level`: The minimum log level that this Logger acts upon. This level is set by default to the ODL `NOTIFICATION:1` value, which maps to the `INFO` Java log level displayed on the Logger Configuration page in Application Server Control Console.

    You can set this value to either a Java logging level (`FINE`) or an ODL Message Type:Log Level (`TRACE:1`).
  - `useParentHandlers`: Indicates whether or not the Logger should use its parent Handlers. Because this value is set to `false` by default, the `oracle` logger does not inherit the log level set for its parent, the `root` logger.
  - `<handler>`: The name of the Handler to use. Do not modify this value.

The following example sets the default log level to `FINEST` by specifying `TRACE:32` as the ODL Message Type:Log Level.

```
<logging_configuration>
  <log_handlers>
```

```
    <log_handler name='oc4j-handler'
        class='oracle.core.ojdl.logging.ODLHandlerFactory'>
      <property name='path' value='%ORACLE_HOME%/j2ee/%OPMN_PROC_TYPE%/log/
          %OPMN_PROC_TYPE%_%OPMN_PROC_SET%_%OPMN_PROC_INDEX%/oc4j'/>
      <property name='maxFileSize' value='10485760'/>
      <property name='maxLogSize' value='104857600'/>
    </log_handler>
  </log_handlers>
  <loggers>
    <logger name='oracle' level='TRACE:32' useParentHandlers='false'>
      <handler name='oc4j-handler'/>
    </logger>
  </loggers>
</logging_configuration>
```

11

Task Manager and Thread Pool Configuration

This chapter provides guidelines for configuring the task manager and thread pool management features for an OC4J instance. It contains the following sections:

- Configuring the OC4J Task Manager
- Using Thread Pools

Configuring the OC4J Task Manager

The *task manager* is a background process that executes all pending tasks, such as timing out HTTP sessions and checking for changed configuration files. By default, it executes every second (1000 milliseconds).

The interval at which the task manager executes is specified in milliseconds in the `taskmanager-granularity` attribute of the `<application-server>` element in the `server.xml` configuration file. This is an OC4J container-level parameter. The default is 1000 milliseconds.

For example, the following entry in `server.xml` configures the task manager to execute every minute (60000 milliseconds):

```
<application-server ...  taskmanager-granularity="60000" ...>
```

Note that you must restart OC4J after making modifications to `server.xml`.

Note: You can also set this parameter through the `granularity` attribute in the `TaskManager` MBean, which is accessible through the JMX Browser in the Application Server Control Console.

See Chapter 12, "Using MBeans in OC4J" for details on accessing and using MBeans to manage OC4J processes.

Using Thread Pools

Thread pools create and store threads for use and re-use by an OC4J process. Re-using existing threads rather than creating new threads on demand improves performance and reduces the burden on the JVM and underlying operating system.

This section covers the following topics:

- Using the Default Thread Pool Configuration
- Managing Thread Pool Configurations

Using the Default Thread Pool Configuration

By default, a single thread pool is created at OC4J startup. New threads are created and added to the pool on an as-needed basis. As each thread is released, it is returned to the pool to remain idle until it is needed by a new request.

There is no limit on the number of threads that can be created within the pool in this configuration. Idle threads in the pool are re-used before a new thread is spawned, unless the number of requests exceeds the number of available threads. After 10 minutes of inactivity, idle threads are automatically destroyed.

This default configuration should be sufficient for most OC4J usage scenarios.

Managing Thread Pool Configurations

Note: Configuring thread pools or modifying the default configuration should be considered expert-mode tasks. It is strongly recommended that the default single thread pool configuration be used.

You can optionally enforce limits on the default thread pool created at OC4J startup through the following:

- Adding the `<global-thread-pool>` element to `server.xml`. (This element is not included by default.) This element includes `min`, `max`, `queue`, and `keepAlive` attributes.
- Updating the attributes of the `ApplicationServerThreadPool` MBean, which is accessible through the System MBean Browser in Application Server Control Console. See "Using the System MBean Browser" on page 12-5 for details on accessing and using MBeans to manage OC4J.

Alternatively, you can create two different thread pools, effectively dividing different types of threads among the pools:

- A *worker thread pool* containing worker threads used in processing RMI, HTTP and AJP requests, as well as MDB listener threads. These are process-intensive and use database resources.
- A *connection thread pool* containing threads such as listener threads, JDBC connection threads, and RMI server connection threads, and background threads. These threads are typically not process intensive.

To create these two separate thread pools, you must configure the `min`, `max`, `queue`, and `keepAlive` attributes for the worker thread pool and the `cx-min`, `cx-max`, `cx-queue`, and `cx-keepAlive` attributes for the connection thread pool. All of these attributes must be configured if creating pools; otherwise you will see the following error message:

```
Error initializing server: Invalid Thread Pool parameter: null
```

See Table 11–1 on page 11-3 for descriptions of the attributes of `<global-thread-pool>`.

The following example initializes two thread pools for the OC4J process. Each contains a minimum of 10 threads and maximum of 100 threads. The number of requests outstanding in each queue can be 200 requests. Also, idle threads are kept alive for 700 seconds. The thread pool information is printed at startup.

```
<application-server ...>
 ...
 <global-thread-pool min="10" max="100" queue="200" keepAlive="700000"
  cx-min="10" cx-max="100" cx-queue="200" cx-keepAlive="700000" debug="true"/>
 ...
</application-server>
```

Table 11–1 below describes the attributes of the `<global-thread-pool>` element, which also correspond to attributes of the `ApplicationServerThreadPool` MBean. Note that the `<global-thread-pool>` element is not included in `server.xml` by default.

Table 11–1 Attributes of <global-thread-pool>

Attribute	Description
`min`	The minimum number of threads to create in the pool. By default, a minimum number of threads are pre-allocated and placed in the thread pool when the container starts. If you add the `<global-thread-pool>` element to `server.xml`, the default value is set to `20`. To disable the thread pool, set this value to `0`.
`max`	The maximum number of threads that can be created in the pool. New threads are spawned if the maximum size is not reached and if there are no idle threads. Idle threads are used first before a new thread is spawned. Note if `<global-thread-pool>` is added to `server.xml`, the default is set to `40`. If this element is not specified, there is no default value.
`queue`	The maximum number of requests that can be kept in the queue. The default is `80`.
`keepAlive`	The length of time, in milliseconds, to keep a thread alive (idle) while waiting for a new request. After the timeout is reached, the thread is destroyed. To never destroy threads, set to `-1`. The default is `600000` milliseconds (10 minutes), which is also the minimum value allowed if not `-1`.
`cx-min`	The minimum number of threads to create in the connection thread pool. The minimum value that can be specified is `1`.
`cx-max`	The maximum number of threads that can be created in the connection pool. The default is `40`.
`cx-queue`	The maximum number of threads that can be kept in the queue in the connection pool. The default is `80`.
`cx-keepAlive`	The length of time, in milliseconds, to keep a thread alive (idle) while waiting for a new request. After the timeout is reached, the thread is destroyed. To never destroy threads, set to `-1`. The default is `600000` milliseconds (10 minutes), which is also the minimum value allowed if not `-1`.
`debug`	If `true`, prints the application server thread pool information to the console at startup. The default is `false`.

A *work management thread pool* containing worker threads used by resource adapters, such as the JMS connector, is created within the OC4J process if needed by a resource adapter deployed to the OC4J instance.

This pool is managed in either of the following ways:

- Adding the `<work-manager-thread-pool>` element to `server.xml`. (This element is not included by default.)
- Updating the attributes of the `WorkManagerThreadPool` MBean, which is accessible through the System MBean Browser in Application Server Control Console. See "Using the System MBean Browser" on page 12-5 for details on accessing and using MBeans to manage OC4J.

The following `server.xml` entry configures the work management thread pool to create a minimum of 10 threads and print thread pool information to the console:

```
<application-server ...>
 ...
 <work-manager-thread-pool min="10" debug="true"/>
 ...
</application-server>
```

Note that the work management thread pool is completely independent of the other two pools; the worker and connection thread pool attributes do not have to be configured in order to activate this pool.

Table 11–2 below describes the attributes of the `<work-manager-thread-pool>` element, which also correspond to attributes of the `WorkManagerThreadPool` MBean. Note that the `<global-thread-pool>` element is not included in `server.xml` by default.

Table 11–2 Attributes of <work-manager-thread-pool>

Attribute	Description
`min`	The minimum number of threads to create in the work management pool. To disable the thread pool, set this value to `0`.
`max`	The maximum number of threads that can be created in the work management thread pool. The default is `40`.
`queue`	The maximum number of threads that can be kept in the queue in the work management pool. The default is `0`, which means that no queue is maintained to handle a sudden burst of work requests.
`keepAlive`	The length of time, in milliseconds, to keep a thread alive (idle) while waiting for a new request. After the timeout is reached, the thread is destroyed. To never destroy threads, set to `-1`. The default is `600000` milliseconds (10 minutes), which is also the minimum value allowed if not `-1`.
`debug`	If `true`, prints the application server work management thread pool information to the console at startup. The default is `false`.

Additional notes on thread pool configuration:

- The `queue` attributes should be at least twice the size of the maximum number of threads.

- The minimum and maximum number of worker threads should be a multiple of the number of CPUs installed on your machine. However, this number should be small; the more threads you have, the more burden you put on the operating system and the garbage collector.
- The `cx-min` and `cx-max` attributes are relative to the number of the physical connections you have at any point in time. The `cx-queue` handles bursts in connection traffic.

12

Using MBeans in OC4J

This chapter describes how the system MBeans provided with OC4J can be used to manage deployed applications, services and other resources within an OC4J instance. It includes the following topics:

- MBeans and Java Management Extensions (JMX) Support in OC4J
- Using the System MBean Browser
- Using JMX Notifications

MBeans and Java Management Extensions (JMX) Support in OC4J

OC4J provides support for the *Java Management Extensions (JMX) 1.2* specification, which allows standard interfaces to be created for managing resources, such as services, applications and resources, in a J2EE environment.

The Oracle Enterprise Manager 10*g* Application Server Control Console user interface is built on a JMX-compliant client that can be used to completely manage and monitor an OC4J instance. The JMX functionality provided through Application Server Control Console is enabled through Java components known as *MBeans*, which are discussed in the next section.

JMX manageable resources within OC4J include:

- The OC4J server
- Applications and Web modules running within an OC4J instance
- J2EE services, such as JTA and JMS
- OC4J processes, such as Task Manager
- Data source and security configuration

This section discusses the following topics:

- What Are MBeans?
- Overview of the Top-Level OC4J System MBeans
- When Do Changes Made Via MBeans Take Effect?
- How Is MBean Data Persisted?

What Are MBeans?

An *MBean*, or *managed bean*, is a Java object that represents a JMX manageable resource. MBeans are defined in the *J2EE Management Specification (JSR-77)*, which is part of the J2EE 1.4 specification as published by Sun Microsystems.

Each manageable resource within OC4J is managed through an instance of the appropriate MBean. For example, an instance of the `J2EESWebSite` MBean is created at OC4J startup to represent each Web site configured within the server.

Each system MBean provided with OC4J exposes a management interface that is accessible through the System MBean Browser. An MBean's interface is comprised of:

- *Attributes*, name/value pairs of any type that the JMX client can get or set remotely. Attributes are analogous to properties set on a JavaBean. For example, the `state` attribute of `J2EEApplication:petstore` MBean indicates whether or not the application is currently running.
- *Operations*, methods that the JMX client can invoke on the MBean. For example, the `stop` operation can be used to stop the `petstore` application and all of its child applications.
- *Notifications* that can be generated broadcast errors or specific events, such as when a new account is created. For example, a notification can be sent to alert you that the `petstore` application has stopped.

As noted earlier, the Application Server Control Console application is built on top of the system MBeans. When you set a property or perform a task in the user interface, you are actually setting an attribute or invoking an operation on an underlying MBean.

To provide you with greater flexibility, Application Server Control Console also provides direct access to the system MBeans provided with OC4J through the *System MBean Browser* component. See "Using the System MBean Browser" on page 12-5 for details on using this management tool.

Overview of the Top-Level OC4J System MBeans

The following table provides an overview of the top-level OC4J system MBeans exposed through the System MBean Browser interface.

Table 12–1 Top-Level OC4J System MBeans

MBean	Description
`J2EEDomain`	Represents a management domain. This is the top-level management object. All other MBeans bound to the domain are visible beneath this node in the System MBean Browser.
`J2EEServer`	Represents a single OC4J instance.
`ClassLoading`	Provides access to all class-loading-related states in an OC4J server instance. Includes an operation to execute the more than 15 built-in queries provided to aid in troubleshooting class-loading issues on a running OC4J instance. This MBean lazily creates instances of the `ClassLoader` MBean, each representing an instantiated classloader.
`EJBCompiler`	Configures the OC4J instance to generate client-side IIOP stubs during EJB deployment. Also used to specify the compiler to use for compiling EJBs.

Table 12–1 (Cont.) Top-Level OC4J System MBeans

MBean	Description
J2EEApplication	Represents a J2EE application deployed into the OC4J instance. Additional MBean instances are visible as child nodes representing the various components of the application: ▪ OC4JWebModule: Represents the properties set through the OC4J-specific orion-web.xml deployment descriptor generated for a Web module deployed as part of the J2EE application. ▪ WebModule: Represents the properties set through the J2EE web.xml deployment descriptor packaged with a WAR file. Instances of the JSP and Servlet MBeans are created for active JSPs and servlets within the Web module.
J2EELogging	Represents a Java Logger component defined in the j2ee-logging.xml file. For an overview of the Java logging framework, including log levels, visit Sun's site at http://java.sun.com/j2se/1.4.2/docs/guide/util/logging/overview.html.
J2EEWebSite	Represents a Web site defined within the OC4J server. See Chapter 13, "Managing Web Sites in OC4J" for details on Web site configuration.
JDBCDriver	Represents a specific JDBC driver.
JMSAdministratorResource	Represents the OC4J JMS server used by the OC4J instance. Includes operations for managing the OC4J JMS server and JMS connection factories, as well as adding/removing destinations.
JMSResource	Displays statistics on messages (by type), active handlers and active connections from the JMS server. Child MBeans contain statistics on connection, destination and durable subscriber resources.
JNDINamespace	Returns an XML document containing all JNDI bindings for all applications deployed into the OC4J instance.
JNDIResource	Returns all JNDI bindings for a specific application.
JSPConfig	Configures the OC4J JSP container. See the *Oracle Containers for J2EE Support for JavaServer Pages Developer's Guide* for documentation of the various configuration values. Note that any changes made to MBean attributes require an OC4J server restart to take effect.
JTAResource	Represents a transaction manager instance. Note that invoking the configureCoordinator operation on this MBean requires an OC4J server restart for the new two-phase commit-coordinator configuration to take effect.
JVM	Describes a Java Virtual Machine that an OC4J instance is running within. Includes an operation to get/set system properties and force garbage collection to start.
SecurityProvider	Used to manage security for a specific application. Note that a restart of the corresponding application or the OC4J server is required for some attributes and operations to take effect.
TaskManager	Describes an OC4J task manager instance. This MBean can be used to set task manager granularity.
ThreadPool	Represents a single instantiated thread pool. Use to set the maximum and minimum number of threads in the pool.
TimerService	Represents an instance of the EJB timer. See the *Oracle Containers for J2EE Enterprise JavaBeans Developer's Guide* for details.

When Do Changes Made Via MBeans Take Effect?

Changes can be made to a managed component via an MBean while the component is either stopped or running.

In general, changes made to a managed component - values set in an attribute or the results of an operation - are available immediately in the OC4J runtime.

In some cases, however, new attribute values or operation results will require a restart - of the OC4J server, of the affected application or even of the MBean - before becoming available in the OC4J runtime. In these cases, the MBean and the Application Server Control Console will display the "new" value; however, the "old" value will continue to be used in the OC4J runtime until the required restart is completed.

For example, suppose you change the value of the `timeout` attribute of the `JSPConfig` MBean from `30` to `15`. The new value of `15` will be displayed both in the MBean and in the JSP Container Properties page in Application Server Control Console. However, because all changes to `JSPConfig` attributes require a restart of the OC4J server, the old value of `30` will continue to be used until the server is restarted.

If a restart is required, the System MBean Browser displays a Required Restart property noting the required actions. Table 12–2 below lists the values for this property.

Table 12–2 Required Restart Property Values

Value	Impact
OC4J Restart	Indicates that the OC4J server instance must be restarted.
Application Restart	Indicates that the J2EE application under which the MBean is registered must be restarted. MBeans that belong to this category are displayed under the `J2EEApplication` node in the navigation pane to the left of the console.
MBean Restart	Indicates that the affected MBean must be restarted.

Change is managed at the individual attribute/operation level, rather than at the MBean level. This means that an MBean might contain attributes that require a restart before a new value is available in the runtime, and other attributes that become available immediately.

How Is MBean Data Persisted?

Persistent data set via an MBean is written to the appropriate XML configuration file(s). For example, new values set in attributes of the `JSPConfig` MBean are written to the `global-web-application.xml` configuration file.

Whether an MBean persists data is indicated by the Persist Policy property displayed in the System MBean Browser.

Table 12–3 Persist Policy Property Values

Value	Impact
OnUpdate	Any persistent data set on the MBean is written immediately to the appropriate configuration file(s) at the time the attribute change is applied or the operation is invoked.
Never	Data set on the MBean is not persisted but exists only in runtime memory.

Using the System MBean Browser

The System MBean Browser is a component of the Web-based Oracle Enterprise Manager 10*g* Application Server Control Console user interface. The console itself is relatively simple to use. To use this feature:

1. Launch the Application Server Control Console.
2. Click the **Administration** link.
3. Click **System MBean Browser**.
4. Specific MBean instances are accessed through the navigation pane to the left of the console. Expand a node in the navigation pane and drill down to the MBean you wish to access.
5. Click the **Attributes** tab in the right-hand pane to access the selected MBean's attributes. If you modify any attribute values, click the **Apply Changes** button to apply your changes to the OC4J runtime.

   Note: The **Apply Changes** button will be visible only if the browser page contains at least one attribute with a modifiable value.

6. Click the **Operations** tab to access the MBean's operations. After selecting a specific operation, click the **Invoke** button to call it.

Using JMX Notifications

Many of the system MBeans provided with OC4J include the ability to generate notifications triggered by a state change registered by the MBean. The following section describes how to subscribe to and view MBean-generated notifications.

Note that not all MBeans generate notifications.

Subscribing to Notifications

You can subscribe to notifications either through the System MBean Browser or the Notification Subscriptions page.

To subscribe to one or more of an MBean's notifications through the System MBean Browser:

1. Click the **Administration** link in the Application Server Control Console.
2. Click **System MBean Browser**.
3. Specific MBean instances are accessed through the navigation pane to the left of the console. Expand a node in the navigation pane and drill down to the MBean you wish to access.
4. Click the **Notifications** tab in the right-hand pane to access the selected MBean's notifications. Note that if this tab is not present, the MBean does not generate notifications.
5. Check the **Subscribe** box.
6. Click the **Apply** button.

To subscribe to notifications generated by multiple MBeans through the Notification Subscriptions page.

1. Click the **Administration** link in the Application Server Control Console.
2. Click the **Notification Subscription** icon. All MBeans that generate notifications are displayed.
3. Check the **Subscribe** box for each notification you wish to subscribe to.
4. Click the **Apply** button.

Using Application-Specific MBeans

Vendor-supplied MBeans deployed with a J2EE application into OC4J can be accessed via the application's "home page" in the Application Server Control Console user interface. Through the user interface, you can view and set attributes and invoke operations on application-specific MBeans, just as you can with the OC4J system MBeans.

1. Click the **Applications** link in the Application Server Control Console.
2. Click the name of the application the MBeans belong to. This opens the "home page" for the application.
3. Click the **Application Defined MBeans** link. The MBeans defined by the application are listed on the page displayed.
4. Click the **Attributes** tab in the right-hand pane to access the selected MBean's attributes. If you modify any attribute values, click the **Apply Changes** button to apply your changes to the OC4J runtime.

   Note: The **Apply Changes** button will only be visible if the browser page contains at least one attribute with a modifiable value.

5. Click the **Operations** tab to access the MBean's operations. After selecting a specific operation, click the **Invoke** button to execute.

13

Managing Web Sites in OC4J

This chapter explains how additional Web sites can be configured in an OC4J standalone environment to provide access to Web applications deployed into the OC4J instance. It also explains how to configure and enable a secure Web site utilizing Secure Socket Layer (SSL) communication between the client and OC4J using HTTPS.

The following sections are included:

- What Is a Web Site in OC4J?
- Configuring Web Site Connection Data
- Creating a New Web Site in OC4J
- Configuring a Secure Web Site in OC4J
- Starting/Stopping Web Sites
- Configuring Web Site Access Logging

What Is a Web Site in OC4J?

In the context of OC4J, Web requests sent to applications deployed to an OC4J instance are received by a *Web site*, a listener configured to accept requests on a specific protocol and port (or range of ports). Every Web module deployed into an OC4J instance must be bound to a Web site through which it will be accessed. This binding is typically performed as part of the application deployment process.

A `default` Web site is created in each OC4J instance upon installation. The configuration for the `default` Web site is defined in a configuration file, `default-web-site.xml`, installed by default in the `ORACLE_HOME/j2ee/home/config` directory. See "Configuring Web Site Connection Data" on page 13-2 to gain an understanding of Web site configuration.

- Standalone OC4J

  In a standalone OC4J configuration, the `default` Web site is configured to receive HTTP requests directly on a specific port, which is `8888` by default. The site can alternatively be configured to receive secure HTTPS requests.

- Single OPMN-managed OC4J instance

  In a single OPMN-managed OC4J installation, the `default` Web site can be similarly configured to receive HTTP or HTTPS requests directly. A specific listener port can be specified in `default-web-site.xml`, or a range of ports can be set in the OPMN configuration file (`opmn.xml`). See "Configuring Web Site Data in OPMN-Managed OC4J Instances" on page 13-2 for details.

- Multiple OPMN-managed OC4J instances

In a cluster of two or more OPMN-managed OC4J instances, the `default` Web site is configured to receive requests forwarded from Oracle HTTP Server (OHS) via the Apache JServ Protocol (AJP).

The site can alternatively be configured to receive secure AJPS requests. A specific listener port can be specified, or a range of ports can be set in the OPMN configuration file. See "Configuring Web Site Data in OPMN-Managed OC4J Instances" on page 13-2 for details on OPMN configuration.

In addition to the `default` site, new Web sites can be configured on each OC4J instance, as needed. (A Web site cannot listen on more than one protocol.) Reasons for creating a new Web site might include:

- Separate management and general Web access

  By default, the Application Server Control Console application is accessed via the `/em` context through the default Web site. However, you can create a new Web site specifically for the Application Server Control Console application to separate management access from general application access, if desired.

- Utilizing secure and nonsecure Web sites

  You can configure the default Web site to utilize SSL to create secure connections, or can create an additional site and bind it to Web applications that require a secure connection.

See "Creating a New Web Site in OC4J" on page 13-5 for instructions on creating and configuring additional Web sites.

Configuring Web Site Connection Data

Key differences exist in how the protocol and listener ports used by a Web site is managed in standalone OC4J versus Oracle Application Server environments.

- Configuring Web Site Data in a Standalone OC4J Installation
- Configuring Web Site Data in OPMN-Managed OC4J Instances

Configuring Web Site Data in a Standalone OC4J Installation

In a standalone OC4J installation, the protocol and listener ports used by a Web site must be explicitly defined in the corresponding `*-web-site.xml` configuration file. See "Creating the Web Site Configuration File" on page 13-6 for a overview of these files.

The `default` Web site is configured to listen for requests received via the HTTP protocol on port `8888` by default.

Configuring Web Site Data in OPMN-Managed OC4J Instances

In an Oracle Application Server installation, in which Oracle Process Manager and Notification Server (OPMN) is used to manage OC4J instances, OPMN can be used to efficiently manage Web site protocol and port configuration.

In this model, the protocol a Web site will use is specified within a `<port>` element defined for the Web site in `opmn.xml`, the OPMN configuration file. A range of listener ports the Web site will use can also be specified within this element.

Note: The `opmnctl` command-line tool provides a command that you can use to update the `<port>` element for a specific Web site defined in the `opmn.xml` file for an OC4J instance.

See "Configuring Web Sites with opmnctl" on page 13-4 for usage details.

When OPMN is started, it selects a port value starting at the bottom of the specified range and increments the value by 1 until a free port is found. Allowing OPMN to select from a range of ports in this manner avoids potential conflicts among OC4J processes.

Note that the protocol and port values specified in `opmn.xml` override any corresponding values set in the corresponding Web site configuration file. Note that using OPMN to manage Web site protocol and port settings is not required in an Oracle Application Server environment; you can opt to not set these values in `opmn.xml` and instead set the values directly in the appropriate Web site configuration file.

The `<port>` element is defined in the `opmn.xml` configuration file, which is located in the *`ORACLE_HOME`*`/opmn/conf` directory. The syntax of the element is as follows:

```
<port id="webSiteName" protocol="http|https|ajp|ajps"
 range="startPort-endPort"/>
```

Table 13–1 below describes the attributes of the `<port>` element.

Table 13–1 Attributes of the <port> Element

Attribute	Description
`id`	Required. Defines the name of the Web site, which is the name of the Web site configuration file minus the `.xml` extension.
protocol	Optional. Specifies the protocol the Web site will receive requests through. Valid values are: ▪ `http` ▪ `https` ▪ `ajp` ▪ `ajps` Note that if either `https` or `ajps` is specified, the value of the secure attribute of the root `<web-site>` element in the `*-web-site.xml` configuration file defined for the Web site will be overridden.
`range`	Optional. Specifies the start and end ports for the range of ports available for assignment by OPMN. The default listener port ranges used are: ▪ HTTP: `8888-8987` ▪ AJP: `12501-12600`

All `<port>` elements defining connection protocols are set in the `<process-type>` element defining the OC4J instance. The `<process-type>` element is a subelement of the `<ias-component>` element, where the `id` attribute equals `OC4J`.

For example, the `<port>` element in the following example configures the `default` Web site on the OC4J `home` instance to listen for AJP requests on ports `12501` through `12600`.

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
   ...
    <port id="default-web-site" protocol="ajp" range="12501-12600"/>
    <port id="rmi" range="12401-12500">
    <port id="jms" range="12601-12700">
    <process-set id="default" numprocs="1"/>
  </process-type>
</ias-component>
```

Note: The `opmn.xml` file must be reloaded for changes made to take effect. Run the following command on the affected node to reload `opmn.xml`:

```
opmnctl reload
```

Note that this command will not affect OPMN-managed components, including OHS, OC4J and deployed applications.

Configuring Web Sites with opmnctl

The OPMN command-line tool, `opmnctl`, provides a `config port` command which allows you to specify, update or delete a Web site configuration defined in `opmn.xml`.

The `opmnctl` tool is installed in the `ORACLE_HOME/opmn/bin` directory on each node. The tool must be run individually on each node, and will update only the `opmn.xml` file on that node.

Inserting or Updating Web Site Configuration Data in opnm.xml

The `update` command sets the specified data in a new or existing `<port>` element. The syntax is as follows:

```
opmnctl config port update ias-component=componentName
 process-type=instanceName portid=webSiteName [range=startPort-endPort]
 [protocol=<http|https|ajp|ajps>]>
```

The following command modifies the `default` Web site for an OC4J instance that is currently configured to listen for HTTP requests to receive and respond to Apache JServ Protocol requests from Oracle HTTP Server.

```
opmnctl config port update ias-component=OC4J process-type=home
   portid=default-web-site protocol=ajp

opmnctl reload
```

Note that the `opmnctl reload` command is invoked to reload the updated `opmn.xml` file into the OC4J runtime.

Deleting Web Site Configuration Data from opnm.xml

The `delete` command removes the `<port>` element defined for the specified Web site. The syntax is as follows:

```
opmnctl config port delete ias-component=componentName
    process-type=instanceName portid=webSiteName
```

For example, the following removes the `<port>` element defined for the `default` Web site from `opmn.xml`:

```
opmnctl config port update ias-component=OC4J process-type=home
    portid=default-web-site

opmnctl reload
```

Table 13–2 below describes the options that can be set on the `opmnctl config port` command line.

Table 13–2 opmnctl config port Options

Option	Description
`ias-component`	Set to `OC4J` to update the OC4J configuration in `opmn.xml`.
`process-type`	Set to the identifier of the OC4J instance to update; for example, `home`. This value matches the value of the `id` attribute in the `<process-type>` subelement of `<ias-component>` in `opmn.xml`.
`portid`	Set to the name of the Web site, which is the name of the Web site configuration file minus the `.xml` extension.
`protocol`	Specifies the protocol the Web site will receive requests through. Valid only for the `update` operation. Valid values are: ■ `http` ■ `https` ■ `ajp` ■ `ajps` Note that if either `https` or `ajps` is specified, the value of the secure attribute of the root `<web-site>` element in the `*-web-site.xml` configuration file defined for the Web site will be overridden.
`range`	Set the start and end ports for the range of ports available for assignment by OPMN. Valid only for the `update` operation. The default port ranges are: ■ HTTP: `8888-8987` ■ AJP: `12501-12600` Note that a single port can be specified instead of a range.

Creating a New Web Site in OC4J

Bringing a new Web site to life in an OC4J instance is essentially a two- or optionally three-step process:

1. Create the XML configuration file for the Web site within the OC4J installed directory structure.
2. Add a reference to the new Web site configuration file in `server.xml`, the OC4J configuration file.
3. For OPMN-managed OC4J instances, add a `<port>` element defining the Web site's protocol and port ranges to `opmn.xml`.

Once these steps are completed, the Web site will be available for binding with applications. The following topics provide details on Web site configuration.

- Creating the Web Site Configuration File
- Referencing the Web Site Configuration File in server.xml
- Defining the Web Site Connection Data in opmn.xml
- Sharing Web Applications Between Web Sites

- Specifying the Cookie Domain

Creating the Web Site Configuration File

> **Note:** This discussion provides instructions on configuring a Web site using XML configuration files. Web sites can also be created in standalone OC4J installations using the `admin.jar` command-line utility. See "Managing Web Sites" on page 7-8 for instructions.

The key information defined in a Web site configuration file includes the following:

- The Web context for each application bound to the site, which is appended to the URL used to access the site (for example, `/em`).
- The protocol the site uses. Note that in an OPMN-managed environment, this value will be overridden by the protocol specified in `opmn.xml`.
- The port the site listens on. In an OPMN-managed environment, this value will be overridden by the port range specified in `opmn.xml`.
- The location of the access log file, which tracks user access to the site.

The most straightforward way to create a new configuration file is to make a copy of the default Web site configuration file, `default-web-site.xml`, which is located in the `ORACLE_HOMEj2ee/home/config` directory. Name the file according to the following convention:

webSiteName`-web-site.xml`

The typical configuration file includes a root `<web-site>` element containing attributes that specify the following:

- `port`: The Web site listener port.
- `display-name`: The for-display name of the Web site.
- `virtual-hosts`: Any additional domains bound to this Web site.

The `<web-site>` element also typically contains the following subelements:

- A `<default-web-app>` element defining the Web application accessed by default through the Web site. When a single application is bound to the Web site - such as Application Server Control Console - specify the application within this element.
- One or more `<web-app>` sub-elements for each Web module bound to the Web site. These elements are added by OC4J when each application is bound to the Web site; however, they can be added to the file manually if desired. At a minimum, each `<web-app>` element has the following:
  - An `application` attribute to specify the name of the J2EE application to which the Web module belongs (the same as the EAR file name without the `.ear` extension)
  - A `name` attribute to specify the name of the Web module (the same as the WAR file name without the `.war` extension)
  - A `root` attribute to specify the context path on this Web site to which the Web module is to be bound

- An <access-log> element specifying the log file that requests sent to the site are logged to

As an example, assume that you will create a configuration file named ascontrol-web-site.xml, which defines a Web site that will be used exclusively to provide management access to Application Server Control Console. The root <web-site> element within this file will contain all of the required configuration data, as shown below:

```
<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="1810"
  display-name="Application Server Control Web Site">
    <default-web-app application="ascontrol" name="ascontrol"/>
    <access-log path="../log/ascontrol-web-access.log" />
</web-site>
```

See the <web-site> element description on page B-15 for details on the structure of this element.

> **Note:** If you are creating a Web site exclusively for use by Application Server Control Console, as illustrated in this example, note that you must also update the **Launch Application Server Control** link on OC4J home page, accessed via *ORACLE_HOME*/j2ee/, with the correct URL.

Referencing the Web Site Configuration File in server.xml

The location of every Web site configuration file must be referenced in a <web-site> element in server.xml, the OC4J configuration file located in the *J2EE_HOME*/config directory. Note that applications will not be able to bind to the Web site unless this declaration exists in server.xml.

Each <web-site> element specifies the path and file name for the corresponding Web site XML file, as in the following sample server.xml entries:

```
<application-server ... >
  <web-site path="./default-web-site.xml" />
  <web-site path="./ascontrol-web-site.xml" />
</application-server>
```

In this example, the locations of all of the Web site configuration files are relative to the location of server.xml.

> **Note:** If OC4J polling is disabled, OC4J must be restarted for changes to server.xml to take effect.

Defining the Web Site Connection Data in opmn.xml

In an Oracle Application Server installation, in which Oracle Process Manager and Notification Server (OPMN) is used to manage OC4J instances, OPMN can be used to efficiently manage Web site protocol and port configuration.

Use the opmnctl config port command to add a new <port> element for the Web site to the OC4J instance definition in opmn.xml. The following example sets the protocol (HTTP) and port (1810) for the ascontrol Web site:

```
opmnctl config port update ias-component="home" id="ascontrol-web-site"
```

```
protocol="http" range="1810"
```

The example command adds the new <port> element to the OC4J home instance definition in the opmn.xml file on the host machine. This OC4J instance is now configured with two Web sites: the default site and the new ascontrol site.

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
   ...
    <port id="default-web-site" protocol="ajp" range="12501-12600"/>
    <port id="ascontrol-web-site" protocol="http" range="1810"/>
    <port id="rmi" range="12401-12500">
    <port id="jms" range="12601-12700">
    <process-set id="default" numprocs="1"/>
  </process-type>
</ias-component>
```

Sharing Web Applications Between Web Sites

Sharing a Web application implies the sharing of everything that makes up the application, including sessions, servlet instances, and context values.

A typical use for this mode is to share a Web application between an HTTP site and an HTTPS site on the same context path - essentially *binding* the application to the two different Web sites. This results in improved performance because only sensitive information is encrypted as needed, rather than requiring that all information in a request be encrypted.

Another benefit is that the cookie, rather than the SSL certificate, is used to track the session. The SSL certificate uses 50 KB to store each certificate when tracking it, which sometimes results in an "out of memory" problem for the session before the session times out. This could possibly make the Web application less secure, but might be necessary to work around issues such as SSL session timeouts not being properly supported in some browsers.

You can set an application as shared by setting the shared attribute of the <web-app> element to true in the *-web-site.xml file defining each Web site the application is bound to. This attribute is false by default.

For example, the sample petstore application is shared between both the default OC4J Web site - which listens on port 8888 - and a new secure Web site listening on port 4443 - by adding or modifying the following <web-app> elements in each Web site configuration file. This configuration will enable the application to accept both HTTP and HTTPS connections.

The <web-app> entry in default-web-site.xml:

```
<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="8888" display-name="OC4J 10g (10.1.3) HTTP Web Site">
    <web-app application="petstore" name="petstore" load-on-startup="true"
      root="/petstore" shared="true"/>
    <access-log path="../log/http-web-access.log" />
</web-site>
```

The similar entry in secure-web-site.xml:

```
<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="4443" secure ="true" display-name="My Secure Web Site">
```

```
  <web-app application="petstore" name="petstore" load-on-startup="true"
   root="/petstore" shared="true"/>
  <access-log path="../log/secure-web-access.log" />
  <ssl-config factory="com.evermind.ssl.JSSESSLServerSocketFactory"
    keystore="../../server.keystore" keystore-password="welcome"
    provider="com.sun.net.ssl.internal.ssl.Provider" />
</web-site>
```

Specifying the Cookie Domain

Note that you can optionally set the *cookie domain* to a specific value. This causes the domain to be set to the specified value at the time a cookie is created, resulting in a cookie that can be sent by a Web browser to any Web site within the domain.

If the domain is not specified, the browser defaults to the domain of the fully qualified server name, such as `site1.acme.com`. In this case, the browser would not be able to forward the cookie to `site2.acme.com`. However, if the cookie domain is explicitly set to `acme.com`, the cookie could be sent to either server.

Set the `cookie-domain` attribute in the `<session-tracking>` element in the J2EE standard `orion-web.xml` file for the application. The cookie-domain attribute contains the DNS domain with at least two components of the domain name provided. For example:

```
<session-tracking cookie-domain=".oracle.com" />
```

> **Note:** If the domain is set to `acme.com`, the cookie will not actually be sent to `acme.com`. In this case, `acme.com` must be redirected at the Web server level to `www.acme.com` to enable cookies to be shared among all subdomains of `acme.com`.
>
> The domain cookies will be redirected to should have at least two dots for `org/com/edu/mil/net` domains and three for other domains; for example, `www.acme.com`.

Configuring a Secure Web Site in OC4J

OC4J supports Secure Socket Layer (SSL) communication between the client and OC4J using HTTPS and AJPS. You can modify the configuration file for the `default` Web site to utilize SSL to create secure connections, or can create an additional site and bind it to Web applications requiring a secure connection.

For details on SSL keys and certificates, see the *Oracle Containers for J2EE Security Guide*.

This section covers the following topics

- Creating the Secure Web Site Configuration File

> **Note:** This discussion provides instructions on configuring a secure Web site using XML configuration files. Secure Web sites can also be created using the `admin.jar` command -line utility. See "Managing Web Sites" on page 7-8 for instructions on creating Web sites using `admin.jar`.

Creating the Secure Web Site Configuration File

Specify the appropriate SSL settings under the <web-site> element, as illustrated in the following example.

```
<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="4443" secure="true" display-name="My Secure Web Site">
  <access-log path="../log/secure-web-access.log" />
  <ssl-config factory="com.evermind.ssl.JSSESSLServerSocketFactory"
    keystore="../../server.keystore" keystore-password="welcome"
    provider="com.sun.net.ssl.internal.ssl.Provider" />
</web-site>
```

Note the additions to <web-site>, shown in **bold**:

- Add a secure attribute with the value set to true. Setting secure="true" specifies that the HTTP protocol is to use an SSL socket.
- Set the port attribute to an available port. The default for SSL ports is 443; in the example above, the port attribute is set to 4443.
- Add the <ssl-config> element. This element is required whenever the secure flag is set to true. This element takes the following attributes and elements:
  - The optional factory attribute is used to specify the third-party SSLSecureSocketFactory implementation to use if the application is not using JSSE.

    The Oracle implementation - com.evermind.ssl.JSSESSLServerSocketFactory - is used by default. (Note that although the default implementation is shown in the example, it is implicit and does not need to be specified.)

    If the application uses a third-party SSLServerSocketFactory implementation, you can use <property> subelements of <ssl-config> to send parameters to the factory.
  - The keystore and keystore-password attributes specify the directory path and password for the keystore. The specified keystore must contain the certificates of any clients that are authorized to connect to OC4J through HTTPS. The value of keystore can indicate either an absolute or relative directory path and includes the file name.
  - The optional provider attribute can be used to specify a security provider to use.

    By default, the Sun Microsystems implementation - com.sun.net.ssl.internal.ssl.Provider - is used. (Note that although the default implementation is shown in the example, it is implicit and does not need to be specified.)
  - One or more <property> elements containing parameters to pass to the SSLSecureSocketFactory. Each element contains a name attribute and a value attribute, enabling you to specify parameters as name/value pairs.

When the Web site configuration file is ready, add a <web-site> element referencing it to server.xml, the OC4J configuration file located in the *J2EE_HOME*/config directory. Note that applications will not be able to bind to the Web site unless this notation exists in server.xml. For example:

```
<application-server ... >
  <web-site path="./default-web-site.xml" />
```

```
  <web-site path="./mycustom-web-site.xml" />
  <web-site path="./secure-web-site.xml" />
</application-server>
```

When configuration is complete, OC4J listens for SSL HTTP requests on one port and non-SSL HTTP requests on another. You can disable either SSL requests or non-SSL requests by commenting out the appropriate `*-web-site.xml` in the `server.xml` configuration file.

```
<!--  <web-site path="./secure-web-site.xml" /> commented out to remove SSL -->
```

For more information about elements and attributes of the `<web-site>`, `<web-app>`, and `<session-tracking>` elements, see the XML Appendix in the *Oracle Containers for J2EE Servlet Developer's Guide.*

Requiring Client Authentication

You can require that clients be authenticated by the server by setting the `needs-client-auth` attribute of the `<ssl-config>` element to `"true"`. For example:

```
<web-site ... secure="true" ... >
 <ssl-config factory="com.evermind.ssl.JSSESSLServerSocketFactory"
   keystore="../../server.keystore" keystore-password="welcome"
   needs-client-auth="true" />
</web-site>
```

This step sets up a mode where OC4J accepts or rejects a client entity for secure communication, depending on its identity. The `needs-client-auth` attribute instructs OC4J to request the client certificate chain upon connection. If the root certificate of the client is recognized, then the client is accepted.

The keystore specified in the `<ssl-config>` element must contain the certificates of any clients that are authorized to connect to OC4J through HTTPS.

Requesting Client Authentication with OC4J

OC4J supports a "client-authentication" mode in which the server explicitly requests authentication from the client before the server will communicate with the client. In this case, the client must have its own certificate. The client authenticates itself by sending a certificate and a certificate chain that ends with a root certificate. OC4J can be configured to accept only root certificates from a specified list in establishing a chain of trust back to the client.

A certificate that OC4J trusts is called a *trust point*. This is the first certificate that OC4J encounters in the chain from the client that matches one in its own keystore. There are three ways to configure trust:

- The client certificate is in the keystore.
- One of the intermediate certificate authority certificates in the client's chain is in the keystore.
- The root certificate authority certificate in the client's chain is in the keystore.

OC4J verifies that the entire certificate chain up to and including the trust point is valid to prevent any forged certificates.

If you request client authentication with the `needs-client-auth` attribute, perform the following:

1. Decide which of the certificates in the client's chain is to be your trust point. Ensure that you either have control of the issue of certificates using this trust point or that you trust the certificate authority as an issuer.
2. Import the intermediate or root certificate in the server keystore as a trust point for authentication of the client certificate.
3. If you do not want OC4J to have access to certain trust points, make sure that these trust points are not in the keystore.
4. Execute the preceding steps to create the client certificate, which includes the intermediate or root certificate installed in the server. If you wish to trust another certificate authority, obtain a certificate from that authority.
5. Save the certificate in a file on the client.
6. Provide the certificate on the client initiation of the HTTPS connection.
   a. If the client is a browser, set the certificate in the client browser security area.
   b. If the client is a Java client, you must programmatically present the client certificate and the certificate chain when initiating the HTTPS connection.

Starting/Stopping Web Sites

A Web site is available by default once it has been configured on an OC4J instance. However, Application Server Control Console provides the ability to stop and start inidividual Web sites through the **Administration>J2EE Websites** pages. These pages also display the configuration for each Web site, and provide access to the Web modules bound to each site.

Note: Note that the `ascontrol-web-site`, used by Application Server Control Console, cannot be stopped via the user interface.

---

1. Click the **Administration** link in the Application Server Control Console.
2. Click the **J2EE Websites** icon under **Administration Tasks>Properties**. The Web sites configured on the OC4J server instance are listed on the page displayed.
3. Click the name of the desired Web site.

Configuring Web Site Access Logging

OC4J provides the ability to generate an *access log* for each Web site, which records requests submitted by clients to the Web site.

Access logs can be generated as either text-based log files or as Oracle Diagnostic Logging (ODL) files, which are generated in XML format that is viewable through Application Server Control Console. Only one type of access logging may be configured for a Web site.

Access logging is configured for a Web site in the Web site configuration file (`*-web-site.xml`) using either the `<access-log>` or `<odl-access-log>` element. If neither element is included in the configuration file, access logs are not generated for the Web site.

This section covers the following topics:

- Configuring Text-Based Access Logging

- Viewing Text Access Log Files
- Configuring ODL Access Logging
- Viewing ODL Access Log Files
- Enabling/Disabling Access Logging for a Web Module/Application

Configuring Text-Based Access Logging

Text-based access logging is configured through the `<access-log>` subelement of the root `<web-site>` element in the corresponding Web site's configuration file (`*-web-site.xml`).

> **Note:** It is important to monitor text-based access log files, as this logging format does not support log rotation. If left unchecked, access log files will continue to grow and can overrun the disk.

This `<access-log>` element has the following attributes:

- `path`: Specifies the path and filename of the access log. This is the only required attribute; specifying it alone will cause access logs to be generated.

  The path must be relative to the `j2ee/home/config` directory to enable the log to be viewed through Application Server Control Console, as illustrated by the following entry in `default-web-site.xml`:

  ```
  <access-log path="../log/default-web-access.log" />
  ```

- `format`: Specifies one or more of several supported variables that result in information being prepended to log entries. Supported variables are `$time`, `$request`, `$ip`, `$host`, `$path`, `$size`, `$method`, `$protocol`, `$user`, `$status`, `$referer`, `$agent`, `$cookie`, `$header`, and `$mime`. Between variables, you can type in any separator characters that you want to appear between values in the log message. The default setting is as follows:

  ```
  "$ip - $user - [$time] '$request' $status $size"
  ```

  This default configuration results in log messages such as the following, with the second message wrapping around to a second line:

  ```
  148.87.1.180 - - [17/Nov/2004:10:23:18 -0800] 'GET / HTTP/1.1' 200 2929
  148.87.1.180 - - [17/Nov/2004:10:23:53 -0800] 'GET
  /webservices/statefulTest HTTP/1.1' 200 301
  ```

  In this example, the user is null, the time is in brackets (as specified in the `format` setting), the request is in single-quotes (as specified), and the status and size in the first message are 200 and 2929, respectively.

- `split`: Specifies how often to begin a new access log. Supported values are `"none"` (equivalent to `"never"`, which is the default), `"hour"`, `"day"`, `"week"`, and `"month"`. Note that if `split` is specified, the `suffix` attribute (documented below) can be used to specify timestamp data to append to the file name.

- `suffix`: Specifies timestamp information to append to the base file name of the logs if the `split` attribute is specified.

  The default `suffix` setting is `"-yyyy-MM-dd"`.

  As an example, assume the following `<access-log>` element with `split` specified, using the default `suffix` value:

```
<access-log path="../log/mysite-web-access.log" split="day" />
```

The log file generated will be named as follows:

```
mysite-web-access-2004-11-17.log
```

The format used is that of `java.text.SimpleDateFormat`, and symbols used in `suffix` settings are according to the symbology of that class. Characters are case-sensitive, as described in the `SimpleDateFormat` documentation. For information about `SimpleDateFormat` and the format symbols it uses, refer to the current Sun Microsystems Javadoc at the following location:

http://java.sun.com/j2se/

The following entry in `default-web-site.xml` will generate a file named `default-web-access.log` file:

```
<web-site>
...
  <access-log path="../log/default-web-access.log" />

</web-site>
```

The files will be generated in the following locations, depending on your OC4J installation.

- Standalone OC4J:

  Log files will be generated in *ORACLE_HOME*`/j2ee/home/log/`.

- Oracle Application Server:

  Files will be generated in an OC4J instance- specific directory named *ORACLE_HOME*`/j2ee/`*instance_name*`/application-deployments/log/`*instance_name*`_default_group_1/`.

Viewing Text Access Log Files

Access log text files can be viewed by clicking the **Logs** link in the Web-based Application Server Control Console. ODL log files are identified in the Log Files page by the `.log` extension.

1. Click the **Logs** link at the bottom of any Application Server Control Console page.
2. Expand **OC4J**.
3. Expand **<instanceName>**. The default instance name is `home`.

Configuring ODL Access Logging

In the ODL framework, log files are formatted as XML documents. A key benefit of ODL access logging is that unlike text-based logging, log file rotation is supported.

ODL access logging is configured through the `<odl-access-log>` subelement of the root `<web-site>` element in a Web site's configuration file. This element has the following attributes, all of which are required:

- `path`: The path to the directory where the `log.xml` files for the Web site will be generated.

  The path must be relative to the `*-web-site.xml` configuration file to enable the log files to be viewed through Application Server Control Console.

For easier management, include the name of the Web site in the path.

- `max-file-size`: The maximum size, in kilobytes, that an individual log file is allowed to grow to. When this limit is reached, a new log file is generated.
- `max-directory-size`: Sets the maximum size, in kilobytes, allowed for the log file directory. When this limit is exceeded, log files are purged, beginning with the oldest files.

New files named `log.xml` are generated within the directory specified in the `path` attribute until the maximum directory size is reached. Each log file is equal to or less than the maximum size specified in the attributes.

For example, the following entry in `default-web-site.xml` will cause `log.xml` files to be generated. It will also set log files to a maximum of 1000 KB and the directory maximum to 10,000 KB in a `/default-web-access` directory within `ORACLE_HOME/j2ee/home/log`.

```
<web-site>
...
  <odl-access-log path="../log/default-web-access/" max-file-size="1000"
    max-directory-size="10000" />
</web-site>
```

The files will be generated in the following locations, depending on your OC4J installation.

- Standalone OC4J:

  Log files will be generated in *`ORACLE_HOME`*`/j2ee/home/log/default-web-access`.

- Oracle Application Server:

  Files will be generated in an OC4J instance-specific directory named *`ORACLE_HOME`*`/j2ee/`*`instance_name`*`/application-deployments/log/`*`instance_name`*`_default_group_1/default-web-access`.

Viewing ODL Access Log Files

ODL-formatted log files can be viewed by clicking the **Logs** link in the Web-based Application Server Control Console, allowing administrators to aggregate and view the logging output generated by all components and applications running within OC4J from one centralized location.

ODL log files are identified in the Log Files page by the `.xml` extension.

1. Click the **Logs** link at the bottom of any Application Server Control Console page.
2. Expand **OC4J**.
3. Expand **<instanceName>**. In both standalone OC4J and OAS, the default instance name is `home`.
4. Expand the **Default Web Site** node.
5. Expand **Diagnostic Message Logs**.

Enabling/Disabling Access Logging for a Web Module/Application

If either the `<access-log>` or `<odl-access-log>` element is defined in a Web site configuration file, access logging is enabled by default for all Web modules within applications bound to the Web site.

However, it is possible to disable access logging for a specific module by setting the `access-log` attribute of the application-specific `<web-app>` element in the configuration file to `false`. This may be desirable in situations where a Web module submits such a massive number of requests that text-based access log files will quickly become bloated.

For example, the following entry in `default-web-site.xml` disables access logging for the `default` application's DMS Web component, but leaves text-based access logging for the `admin_web` module enabled:

```
<web-site ...>
  <web-app application="default" name="dms0" root="/dmsoc4j" access-log="false" />
  <web-app application="default" name="admin_web" root="/adminoc4j" />
  <access-log path="../log/http-web-access.log" />
</web-site>
```

14

Registering DTDs and XSDs with OC4J

This chapter describes the process for registering new entities - specifically any vendor-specific DTDs and XSDs used to define the format of XML deployment descriptors - within OC4J, which is required if XML file validation will be performed. It contains the following topics:

- Why Do DTDs/XSDs Have to Be Registered?
- Registering a DTD or XSD

Why Do DTDs/XSDs Have to Be Registered?

OC4J provides the ability to validate XML deployment descriptors at the time the files are read. This feature is enabled by passing the `-validateXML` argument on the `oc4j.jar` command line at OC4J startup. See Chapter 4, "OC4J Runtime Configuration" for details on command-line options.

Validation requires that the DTD or XSD defining an XML document be registered with the OC4J server. If this entity is not registered, XML validation may not occur.

When an XML document is read, the parser passes one or more keys identifying the DTD or XSD declared in the document to an OC4J component known as the *Entity Resolver*. The Entity Resolver resolves the location of the registered entity and returns it to the parser, enabling the XML document to be validated.

Two types of keys are used to reference an entity: A *public identifier* and a *system identifier*, both of which are declared in the XML document. Consider the following declaration in `orion-application.xml`, which references the DTD used to define the file format:

```
<!DOCTYPE orion-application
PUBLIC "-//Evermind//DTD J2EE Application runtime 1.2//EN"
"http://xmlns.oracle.com/ias/dtds/orion-application.dtd">
```

- The *public identifier* is the string `"-//Evermind//DTD J2EE Application runtime 1.2//EN"`
- The *system identifier* is the URL `"http://xmlns.oracle.com/ias/dtds/orion-application.dtd"`

To enable the Entity Resolver to locate the entity, one or both of these identifiers must be registered with OC4J through entries in the `entity-resolver-config.xml` file. The entity's location must also be specified in this file.

Note that by default, `entity-resolver-config.xml` already contains registration entries for the standard J2EE DTDs and XSDs, as well as for all OC4J-specific XSDs. As such, you are only required to add entries for non-J2EE or non-OC4J entities.

Registering a DTD or XSD

To register a DTD or XSD with OC4J, you must add it to the `entity-resolver-config.xml` file, which is located in the `ORACLE_HOME/j2ee/home/config` directory on the OC4J host machine.

Each entity is declared in an `<entity>` element, which includes the following subelements:

- `<description>`: Contains an optional description of the entity.
- `<public-id>`: Contains the entity's public identifier.
- `<system-id>`: Contains the entity's system identifier.

  Either `<public-id>` or `<system-id>` must be specified; however, you are not required to specify both.
- `<location>`: Points to the entity's location. The location can be either the fully qualified path to the entity or a URL that can be resolved locally.

The following `<entity>` element will register `acme-web.dtd` with OC4J. Both the public and system identifiers, which are declared in the `<!DOCTYPE>` element within an XML document, are registered.

```
<entity>
  <description>acme-web-2_0.dtd</description>
  <public-id>-//Acme//Acme web Descriptor 2.0//EN</public-id>
  <system-id>http://xmlns.acme.com/dtd/acme-web-2_0.dtd</system-id>
  <location>META-INF/acme-web-2_0.dtd</location>
</entity>
```

The next example will register `acme-application.xsd` with OC4J. The system identifier is declared in either the `xsi:schemaLocation` or the `xsi:noNamespaceSchemaLocation` attributes of the root element within an XML document.

```
<entity>
  <description>acme-application-1_0.xsd</description>
  <public-id />
  <system-id>http://xmlns.acme.com/schema/acme-application-1_0.xsd</system-id>
  <location>META-INF/acme-application-1_0.xsd</location>
</entity>
```

Note: The OC4J server must be restarted after you make changes to `entity-resolver-config.xml`.

A

Troubleshooting OC4J

This appendix describes common problems that you may encounter when using OC4J and explains how to resolve them. It includes the following topics:

- Problems and Solutions
- Need More Help?

Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

- java.lang.OutOfMemory Errors
- Application Performance Impacted by Garbage Collection Pauses
- Invalid or Unneeded Library Elements Degrading Performance
- ClassCastExceptions and ClassNotFound Errors
- OC4J Fails to Start: Unable to Find Java Compiler
- Error Occurs When Clustering an Application
- Error When Downgrading from JDK 5.0 to JDK 1.4.2
- OC4J Hangs When Starting Applications in Oracle Application Server

java.lang.OutOfMemory Errors

Problem

"Out of memory" errors indicate that the heap size of the Java instance is lower than the memory required by applications running within OC4J.

Solution

Increase the heap size for the OC4J process to the desired amount of memory at OC4J startup:

```
java -Xms512m -Xmx512m -jar oc4j.jar
```

If running in an OPMN managed environment these JVM settings are defined within a `<data id="java-options">` tag the `opmn.xml` configuration file. For example:

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
```

```
      <category id="start-parameters">
        <data id="java-options" value="-Xms512m -Xmx512m -Djava.awt.headless=true
          -Dhttp.webdir.enable=false"/>
        ...
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

If running under Unix/Linux, verify that `ulimit` settings allow the JVM process to allocate this much memory.

Application Performance Impacted by Garbage Collection Pauses

Problem

An application running on OC4J appears unresponsive, with simple requests experiencing noticeable delays. The cause is that the JVM has crossed the low memory threshold and is running a full garbage collection to free up memory.

Solution

Consider using the *incremental low pause collector*, which avoids long major garbage collection pauses by doing portions of the major collection work at each minor collection. This collector (also known as the *train collector*) collects portions of the tenured generation - a memory pool holding objects that are typically collected in a major collection - at each minor collection. The result is shorter pauses spread over many minor collections.

Note that the incremental collector is even slower than the default tenured generation collector when considering overall throughput.

To use the incremental collector, the `-Xincgc` option must be passed in on the Java command line at application startup. Set the initial and maximum size of the young generation (object pool) to the same value using the `XX:NewSize` and `-XX:MaxNewSize` options. Set the initial and the maximum Java heap sizes to the same value using the `-Xms` and `-Xmx` options.

For example, to use this collector with a server with 1 GB of physical memory:

```
java -server -Xincgc -XX:NewSize=64m -XX:MaxNewSize=64m -Xms512m -Xmx512m
```

For more information on garbage collection tuning, read *"Tuning Garbage Collection with the 1.4.2 Java Virtual Machine"* which is available at `http://java.sun.com/docs/hotspot/gc1.4.2/`

Invalid or Unneeded Library Elements Degrading Performance

Problem

If the OC4J process memory is growing consistently during program execution, then you may have references to invalid symbolic links in your global `application.xml` file. This problem is usually characterized by a growth in the C heap and not a growth in Java object memory, as one would see with a more traditional Java object memory leak. OC4J loads all resources using the links in the `application.xml` file. If these links are invalid, then the C heap continues to grow, causing OC4J to run out of memory.

Solution

Ensure that all symbolic links are valid, and restart OC4J.

In addition, keep the number of JAR files OC4J is configured to load to a minimum. Eliminate all unused JAR files from the configuration and from the directories OC4J is configured to search. OC4J searches all JAR files for classes and resources, thereby causing the file cache to use extra memory and processor time.

ClassCastExceptions and ClassNotFound Errors

Problem

Most class loading errors are related to class visibility—either too much or not enough. Collisions between classes packaged in multiple JARs or inherited by default from parent applications can be a problem.

Solution

Chapter 3: Utilizing the OC4J Class Loading Framework in the *Oracle Containers for J2EE Developer's Guide* contains detailed documentation on avoiding and troubleshooting class loading-related issues. It also explains how you can use shared libraries to avoid many of these issues within OC4J.

OC4J Fails to Start: Unable to Find Java Compiler

Problem

An error similar to the one below is seen at OC4J startup:

```
05/10/28 13:58:49 Error initializing server: Error initializing ejb-modules:
Error generating wrappers for file:/C:/oc4j/j2ee/home/applications/admin_ejb.jar:
javac.exe not found under <directory>, please use a valid jdk or specify the
location of your java compiler in server.xml using the <java-compiler .../> tag
```

Solution

The error indicates that OC4J is unable to locate the required JDK. To resolve this issue, start OC4J from the `javac.exe` location on the command line. This will set the location of the JDK.

For example:

```
C:\ORACLE_HOME\j2ee\home\C:\<jdk>\bin\java -jar oc4j.jar
```

Error Occurs When Clustering an Application

Problem

The following error is thrown when clustering is configured for an application:

```
WARNING: The service implementation <classname> does not implement
java.io.Serializable. *This class is not suitable for clustered environments*
indicated by recoverable=true.
```

Solution

This error indicates that the class is not serializable, and therefore cannot utilize the OC4J replication framework.

Error When Downgrading from JDK 5.0 to JDK 1.4.2

Problem

The following error occurs when configuring an OPMN-managed OC4J instance installed as a component of Oracle Application Server, which uses the JDK 5.0 by default, to use the JDK 1.4.2.

```
oracle.oc4j.loader.util.AnnotatedLinkageError:
MBeanServerEjbHome_StatefulSessionHomeWrapper1 (Unsupported major.minor
version 49.0)
```

Solution

An OPMN-managed OC4J instance installed as a component of Oracle Application Server will use the JDK 5.0 by default. This new version of the JDK is required to utilize EJB 3.0 and offers numerous performance improvements. However, if applications that will be deployed to OC4J require a JDK 1.4.2 release, it may be necessary to "downgrade" to the earlier version.

Before switching from JDK 5.0 to JDK 1.4.2, you must remove all compiled application files from the OC4J instance. To do this:

- Stop the OC4J instance.
- Delete the *ORACLE_HOME*/j2ee/*instance*/application-deployments directory.

  Deleting this directory will cause the application files to be re-compiled when OC4J is re-started with the JDK 1.4.2.

OC4J Hangs When Starting Applications in Oracle Application Server

Problem

In an OPMN-managed environment, OPMN appears to "hang" while trying to start OC4J, resulting in a error similar to the following:

```
ias-component/process-type/process-set:
    OC4J/home/default_group/

   Error
   Process (index=1,uid=2012873812,pid=2988)
   time out while waiting for a managed process to start
```

Solution

An application that requires significant resources—such as an application that attempts to acquire multiple database connections for its various components—can cause OC4J to fail to start. You can manage this by specifying the maximum amount of time to allow applications to start in the `<start-timeout>` element defined for the OC4J instance in `opmn.xml`. After this value is reached, the application will not be started. This value will be applied to all applications deployed to the instance.

The following example increases the start time value to 1000 seconds for applications deployed to the `home` OC4J instance:

```
<ias-component id="OC4J">
  ...
  <process-type id="home" module-id="OC4J" status="enabled">
    ...
```

```
    <start timeout="800" retry="2"/>
  </process-type>
</ias-component>
```

Need More Help?

You can search for additional solutions on the following Oracle support-oriented Web sites:

- Oracle Application Server Release Notes, available on the Oracle Technology Network at `http://www.oracle.com/technology/documentation/index.html`
- Oracle MetaLink, available at `http://metalink.oracle.com`

If you still cannot find a solution for the problem you are facing, please log a service request.

B

Configuration Files Used in OC4J

This chapter provides detailed documentation on the XML files used to store configuration data for the OC4J server and J2EE applications and modules deployed into it.

- Overview of the XML Configuration Files Used by OC4J
- Overview of the OC4J Server Configuration File (server.xml)
- Overview of the Web Site Configuration Files (*-web-site.xml)

Overview of the XML Configuration Files Used by OC4J

The configuration data for an OC4J server instance and the applications and modules deployed into it is persisted in a number of XML files. Figure B–1 provides an overview of these XML files and their respective roles.

Note that schemas defining the Oracle-proprietary XML files used by OC4J can be viewed at the following link:

`http://www.oracle.com/technology/oracleas/schema/index.html`

Figure B–1 XML Files Used By OC4J

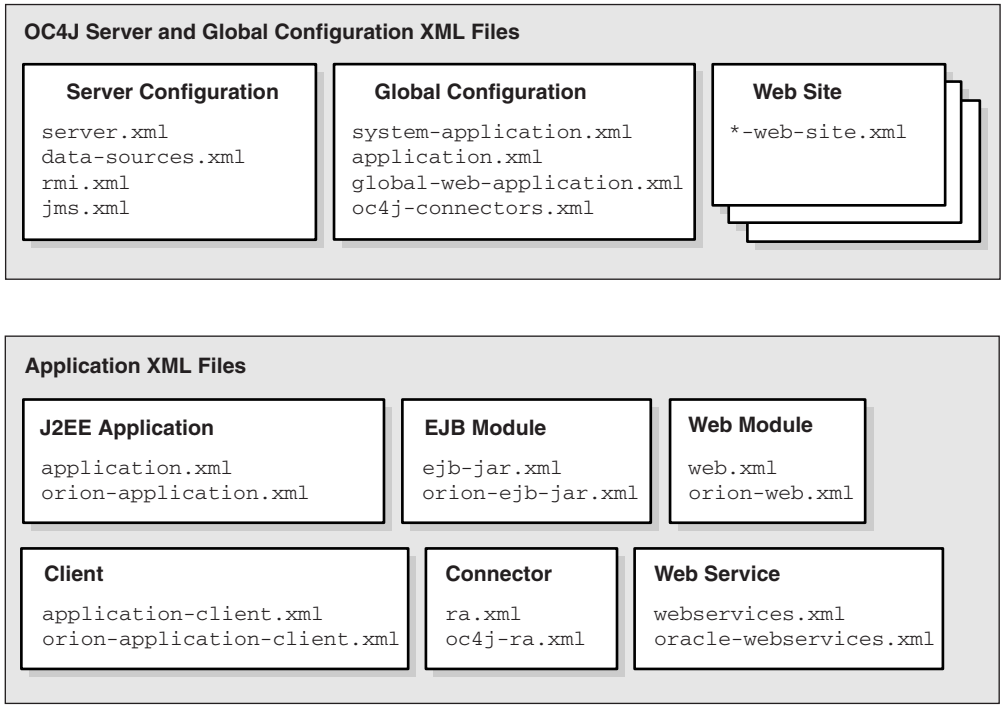


Table B–1 describes the role and function for each OC4J server-level XML file as well as the global configuration files displayed in the preceding figure.

Unless otherwise indicated, all of these files are installed in the `ORACLE_HOME/j2ee/home/config` directory by default.

Table B–1 Server-Level and Global Configuration Files

XML Configuration File	Features/Components
`server.xml`	The OC4J server configuration file. Configures the server and points to the XML files that add to this file, such as `jms.xml` for JMS support. The listing of other XML files enables the services to be configured in separate files, but the `server.xml` file denotes that they be used for the OC4J configuration.
`data-sources.xml`	Contains the OC4J data source configuration for all databases used by applications within OC4J.
`rmi.xml`	Contains OC4J RMI port configuration and RMI tunneling over HTTP.
`jms.xml`	Contains the OC4J JMS configuration for `Destination` topics and queues that are used by JMS and MDBs in OC4J.
`system-application.xml`	Contains the configuration for the `system` application, which is the parent of all other applications installed in the OC4J instance. The file provides configuration data used at OC4J startup, such as data needed to load required shared libraries.

Table B–1 (Cont.) Server-Level and Global Configuration Files

XML Configuration File	Features/Components
application.xml	Contains the configuration for the default application. All user-deployed applications and standalone modules that do not have a designated parent are deployed to this application by default. This file includes common settings that serve as default configuration values applied to deployed applications. Note that this file is completely unrelated to application.xml, the J2EE standard deployment descriptor.
global-web-application.xml	An Oracle-specific file for configuring the servlet and JSP containers within OC4J.
oc4j-connectors.xml	Contains global OC4J-specific configuration data for all standalone resource adapters installed in the OC4J instance.
*-web-site.xml	An OC4J-specific file that contains configuration data for a Web site created within the OC4J instance. It is typically installed in the ORACLE_HOME/j2ee/home/config directory, but may be installed in a different location. The configuration for the default Web site created within each OC4J instance is defined in default-web-site.xml.

Table B–2 describes the role and function for the various application-level XML files displayed in the preceding figure.

Unless otherwise indicated, all of these files are installed in the ORACLE_HOME/j2ee/home/config directory by default.

Table B–2 Application-Level Configuration Files

XML Configuration File	Features/Components
application.xml	The J2EE application standard J2EE application descriptor file. The local application.xml file defines the J2EE EAR file, which contains the J2EE application modules. This file exists within the J2EE application EAR file.
orion-application.xml	The OC4J-specific deployment descriptor, which contains configuration data for a specific deployed application.
web.xml	The J2EE Web application deployment descriptor, used to define the Web application deployment parameters that are included in the WAR file. In addition, you can specify the URL pattern for servlets and JSPs in this file. For example, a servlet is defined in the <servlet> element, and its URL pattern is defined in the <servlet-mapping> element.
orion-web.xml	Extends the standard J2EE descriptor with application-level OC4J-specific configuration data, such as whether or not OC4J features like developer mode or auto-reload of JSPs are enabled.

Table B–2 (Cont.) Application-Level Configuration Files

XML Configuration File	Features/Components
`ejb-jar.xml`	The J2EE EJB module deployment descriptor, included in the EJB JAR file. Defines the specific structural characteristics and dependencies of the Enterprise JavaBeans within a JAR, and provides instructions for the EJB container about how the beans expect to interact with the container.
`orion-ejb-jar.xml`	The OC4J-specific deployment descriptor. Defines OC4J-specific configuration data for all EJBs within an archive, including EJB pool settings, time-out and retry settings, JNDI mappings and finder method specifications. Also includes properties for the TopLink persistence manager.
`application-client.xml`	The J2EE application client configuration file. Describes the EJB modules and other resources used by a J2EE application client packaged in an archive.
`orion-application-client.xml`	Contains OC4J deployment data, including JNDI mappings to an EJB's home interface or to external resources such as a data source, JMS queue or mail session.
`ra.xml`	The J2EE standard deployment descriptor. Contains information on implementation code, configuration properties and security settings for a resource adapter packaged within a RAR file.
`oc4j-ra.xml`	Contains OC4J-specific deployment configuration data for a single resource adapter. This data includes EIS connection information, JNDI name to be used, connection pooling parameters, and resource principal mappings.
`webservices.xml`	The J2EE standard Web services deployment descriptor. Describes a Web service, including WSDL information and JAX-RPC mapping data, for a Web Service application packaged within a WAR file.
`oracle-webservices.xml`	Defines properties used by the OC4J Web services container, such as whether to expose the WSDL file. It also defines endpoint addresses and data specific to EJBs implemented as Web services. The file can be packaged in either a WAR or an EJB JAR containing a Web service.

Overview of the OC4J Server Configuration File (server.xml)

The OC4J configuration file, `server.xml`, is located in the `ORACLE_HOME/j2ee/home/config` directory. It is the starting point for configuration of the OC4J server and all J2EE applications, Web applications and Web sites enabled within the server.

Unless specifically instructed to do so in the OC4J documentation, you should not have to edit `server.xml` manually, as notations are added and updated as needed by OC4J.

The `server.xml` file includes references to the application descriptor of each application within the OC4J instance, either directly or indirectly. In the case of a typical J2EE application, this reference points to the extracted EAR top-level directory

and, therefore, to the `application.xml` file that the EAR file contains. In the case of the OC4J global application, the `server.xml` file points directly to the OC4J global application descriptor.

The `server.xml` file also points to other XML configuration files. For each XML file, the location can be the full path or a path relative to the location of where the `server.xml` file exists. In addition, the name of the XML file can be any name, as long as the contents of the file conform to the appropriate DTD.

- The `<rmi-config>` element denotes the name and location of the `rmi.xml` file.
- The `<jms-config>` element denotes the name and location of the `jms.xml` file.
- The `<global-application>` element denotes the name and location of the global `application.xml` file.
- The `<global-web-app-config>` element denotes the name and location of the `global-web-application.xml` file.
- The `<web-site>` element denotes the name and location of one `*-web-site.xml` file. Since you can have multiple Web sites, you can have multiple `<web-site>` entries.

The `server.xml` file format is described by `application-server-10_1.xsd`, which can be viewed at the following link:

`http://www.oracle.com/technology/oracleas/schema/index.html`

Example of a server.xml File

Below is an example `server.xml`, with `<!-- comments -->` to describe the various sections:

```
<application-server  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
  application-server-10_1.xsd"  application-directory="../applications"
  deployment-directory="../application-deployments"
  connector-directory="../connectors"
  schema-major-version="10" schema-minor-version="0" >
  <!-- Shared library definitions -->
  <shared-library name="global.libraries" version="1.0" library-compatible="true">
    <code-source path="../applib"/>
    <code-source path="../../../sqlj/lib"/>
    <code-source path="../../../lib/dsv2.jar"/>
  </shared-library>
  <shared-library name="global.tag.libraries" version="1.0"
   library-compatible="true">
    <code-source path="../jsp/lib/taglib/standard.jar"/>
  </shared-library>
  <!-- J2EE services -->
  <rmi-config path="./rmi.xml" />
  <sep-config path="./internal-settings.xml" />
  <jms-config path="./jms.xml" />
  <javacache-config path="../../../javacache/admin/javacache.xml" />
  <!-- Logging -->
  <j2ee-logging-config path="./j2ee-logging.xml" />
  <log>
    <file path="../log/server.log" />
  </log>
  <java-compiler name="javac" in-process="false" encoding="ISO8859_1"
   extdirs="c:\sdk\jdk\jre\lib\ext" />
  <!--  Default application configuration -->
```

```
  <global-application name="default" path="application.xml" />
  <!-- Deployed application configuration -->
  <application name="petstore" path="../applications\petstore.ear" start="true" />
  <application name="ascontrol" path="../applications\ascontrol.ear"
   start="true" />
  <!-- Default Web application configuration file -->
  <global-web-app-config path="global-web-application.xml" />
  <!-- Transaction Manager configuration file -->
  <transaction-manager-config path="transaction-manager.xml" />
  <!-- Configuration files for enabled Web sites -->
  <web-site path="./default-web-site.xml" />
</application-server>
```

<application-server>

Required? Required; one only

Child elements:

This is the root element of the OC4J configuration file.

Table B–3 <application-server> Attributes

Name	Description
`application-directory`	Values: string Default: `../applications` The target directory for deployed archives.
`application-auto-deploy-directory`	Values: string Default: n/a The directory into which EAR files can be copied, triggering automatic deployment/redeployment of the application.
`connector-directory`	Values: string Default: ../connectors The target directory for standalone resource adapters.
`deployment-directory`	Values: string Default: ../application-deployments The directory containing the OC4J-specific deployment descriptors and generated files, such as compiled JSP classes and EJB wrapper classes.
`check-for-updates`	Values: `all`|`adminClientOnly`|`none` Default: `adminClientOnly` Enables OC4J polling, which automatically checks for changes made to currently deployed applications and modules, and redeploys any components that have been modified. See the *Oracle Containers for J2EE Deployment Guide* for an explanation of supported values and the impact of each.
`localhostIsAdmin`	Values: Boolean Default: `true` If `true`, allows easier access if the process initiating the administrative operation is a process local to the OC4J host machine.

Table B–3 (Cont.) <application-server> Attributes

Name	Description
taskmanager-granularity	Values: int Default: `1000` The interval at which the task manager performs its duties, specified in milliseconds. The default is every second (1000 milliseconds).

<application>

Parent element: <application-server>

Required? Optional; multiple allowed

Child elements:

Defines a J2EE application deployed into the OC4J instance. The `<application>` element defining an application is added to `server.xml` by OC4J at the time the application is deployed. As such, there is generally no need to manually modify this element.

Table B–4 <application> Attributes

Name	Description
`name`	Values: string Default: `n/a` The application name; typically the same as the EAR file name without the `.ear` extension.
path	Values: string Default: `n/a` The location of the EAR file or the extracted EAR top-level directory. As such, the path indirectly points to the J2EE standard `application.xml` descriptor packaged with the application.
`start`	Values: Boolean Default: `true` If `true`, the application is started with OC4J and is available to serve requests or for configuration through JMX MBeans. If `false`, the application is not started with OC4J, meaning it is not available to serve requests. However, it is available for configuration through JMX.

<code-source>

Parent element: <shared-library>

Required? Required; multiple allowed

Specifies the path to a JAR or ZIP file included in the shared library definition.

Table B–5 <code-source> Attributes

Name	Description
path	Values: string Default: n/a The path to a JAR or ZIP file included in a shared library. Paths may be absolute if outside of the `/shared-lib` directory, or can be relative to the subdirectory containing the JAR files within the `/shared-lib/`*`library_name`* directory. If relative, only the archive file name needs to be supplied as the value of the `path` attribute. You can optionally set `path="*"` to force OC4J to consume all of the archives within the shared library subdirectory.

<execution-order>

Parent element: startup-class, shutdown-class

Required? Optional; one only

Child elements:

Specifies the order of execution for each startup class. Specify an integer that designates the order in which the classes are executed.

<global-application>

Parent element: <application-server>

Required? Required; one only

Child elements:

Specifies the OC4J global application, known as the `default` application. The `name` attribute defines its name; the `path` attribute specifies what to use as the OC4J global application descriptor.

Table B–6 <global-application> Attributes

Name	Description
`name`	Values: string Default: `default` The global application name.
`path`	Values: string Default: `application.xml` The filename and path for the global application descriptor file. The default descriptor is `ORACLE_HOME/j2ee/home/config/application.xml`.

<global-thread-pool>

Parent element: <application-server>

Required? Required; one only

Child elements:

Contains the configuration for a single thread pool within the OC4J process. See "Using Thread Pools" on page 11-1 for details. Note that the `cx-*` attributes are used to configure the second thread pool if two pools are being created.

Table B–7 <global-thread-pool> Attributes

Name	Description
min	Values: string Default: n/a The minimum number of threads that OC4J can simultaneously execute.
max	Values: string Default: n/a The maximum number of threads that OC4J can simultaneously execute.
queue	Values: string Default: n/a The maximum number of requests that can be kept in the queue.
keep-alive	Values: string Default: n/a The length of time, in milliseconds, to keep a thread alive (idle) while waiting for a new request. This timeout designates how long an idle thread remains alive. If the timeout is reached, the thread is destroyed.
cx-max	Values: string Default: n/a The minimum number of connection threads that OC4J can simultaneously execute in the second pool.
cx-min	Values: string Default: n/a The maximum number of connection threads that OC4J can simultaneously execute in the second pool.
cx-queue	Values: string Default: n/a The maximum number of requests that can be kept in the queue in the second pool.

<global-web-app-config>

Parent element: <application-server>

Required? Required; one only

Child elements:

Identifies the configuration file for the OC4J global web application, which by default is the parent of all other Web applications.

The name and root directory path of the default Web application are specified in the global application descriptor, and the default Web application is bound to a Web site through the `default-web-site.xml` file. In OC4J standalone, the default context path for the default Web application is "/".

Table B–8 <global-web-app-config> Attributes

Name	Description
path	Values: string Default: `global-web-application.xml` The filename and path of the global Web application descriptor file. The default descriptor is `ORACLE_HOME/j2ee/home/config/global-web-application.xml`.

<import-shared-library>

Parent element: <shared-library>

Required? Optional; multiple allowed

Identifies a shared library to be imported by a shared library defined in the enclosing `<shared-library>` element. For additional information on configuring and using shared libraries, see the *Oracle Containers for J2EE Developer's Guide*.

Table B–9 <import-shared-library> Attributes

Name	Description
`name`	Values: string Default: required The name of the shared library to import.
version	Values: string Default: required The version number to import.

<init-param>

Parent element: <startup-class>, <shutdown-class>

Required? Optional; multiple allowed

Child elements: <param-name>, <param-value>

Specifies initialization parameters within a `<startup-class>` or `<shutdown-class>` element. Contains key-value pairs, of type `String`, which OC4J takes, which are provided within the input `Hashtable` argument. The names for the key-value pairs must be unique, as JNDI is used to bind each value to its name.

Table B–10 <init-param> Attributes

Name	Description
path	Values: string Default: `global-web-application.xml` The filename and path of the global Web application descriptor file. The default descriptor is `ORACLE_HOME/j2ee/home/config/global-web-application.xml`.

<j2ee-logging-config>

Parent element: <application-server>

Required? Optional; only one allowed

Child elements:

Defines the file to use as the J2EE logging configuration file.

Table B–11 <j2ee-logging-config> Attributes

Name	Description
path	Values: string Default: `./j2ee-logging.xml` The filename and path of the logger configuration file.

<java-compiler>

Parent element: <application-server>

Required? Optional; one only

Child elements:

Specifies configuration parameters for the Java compiler to use to compile EJBs. By default, the `javac` compiler installed with the JDK defined in the `JAVA_HOME` environment variable will be used.

Table B–12 <java-compiler> Attributes

Name	Description
name	Values: string Default: `modern|classic|javac|ojc|jikes` The name of the Java compiler to use.
in-process	Values: Boolean Default: `false` Specifies whether to run the compiler in-process or out-of-process. If set to `false`, a separate JVM process is spawned for the compiler to execute within. This is the default compiler execution mode used by OC4J, as it offers better management of memory resources. If set to `true`, the compiler executes within the same JVM process as OC4J.
`encoding`	Values: string Default: `ISO-8859-1` The source file encoding to use.
bindir	Values: string Default: n/a The absolute path to the directory containing the compiler executable. This attribute does not need to be specified to use the default `javac` compiler.

Table B–12 (Cont.) <java-compiler> Attributes

Name	Description
`extdir`	Values: string Default: `n/a` The compiler extension library location, if applicable.
`debug`	Values: Boolean Default: `false` Set to `true` to generate compilation-time debugging output.

<javacache-config>

Parent element: <application-server>

Required? Optional; only one allowed

Child elements: None

Specifies the path to `javacache.xml`, the Java Object Cache configuration file.

Table B–13 <javacache-config> Attributes

Name	Description
path	Values: string Default: `../../../javacache/admin/javacache.xml` The path to the `javacache.xml` file.

<jms-config>

Parent element: <application-server>

Required? Optional; only one allowed

Child elements:

Specifies the file to use as the OC4J JMS configuration file.

Table B–14 <jms-config> Attributes

Name	Description
path	Values: string Default: `jms.xml` The filename and path of the OC4J JMS configuration file.

<log>

Parent element: <application-server>

Required? Optional; only one allowed

Child elements: <file>

The enclosed <file> element points to the location of the OC4J server log file.

<rmi-config>

Parent element: <application-server>

Required? Optional; only one allowed

Child elements:

Defines the file to use as the OC4J RMI configuration file.

Table B–15 <rmi-config> Attributes

Name	Description
path	Values: string Default: `rmi.xml` The filename and path of the OC4J RMI configuration file.

<shared-library>

Parent element: <application-server>

Required? Optional; multiple allowed

Child elements: <code-source>, <import-shared-library>

Declares a shared library installed within the OC4J instance. For additional information on configuring and using shared libraries, see the *Oracle Containers for J2EE Developer's Guide*.

Table B–16 <shared-library> Attributes

Name	Description
`name`	Values: string Default: required The name of the shared library directory created within the `/shared-lib` directory.
version	Values: string Default: required The version number that serves as the name of the subdirectory containing the shared library's archive files in the `/shared-lib/`*`library_name`* directory.
`library-compatible`	Values: Boolean Default: `false` This attribute is intended for internal use only.

<shutdown-class>

Parent element: shutdown-classes

Required? Optional; multiple allowed

Child elements: execution-order, init-param

Defines a shutdown class to execute before OC4J terminates within the `<startup-classes>` element.

Table B–17 <shutdown-class> Attributes

Name	Description
`classname`	Values: string Default: required The name of the class that implements the `oracle.j2ee.server.OC4JShutdown` interface.

<startup-class>

Parent element: startup-classes

Required? Optional; multiple allowed

Child elements: execution-order, init-param

Defines a startup class to execute on OC4J initialization within the `<startup-classes>` element.

Table B–18 <startup-class> Attributes

Name	Description
`classname`	Values: string Default: required The name of the class that implements the `oracle.j2ee.server.OC4JStartup` interface.
failure-is-fatal	Values: Boolean Default: `false` If `true`, OC4J logs an exception and exits when an exception is thrown. If `false`, OC4J logs the exception and continues.

<transaction-manager-config>

Parent element: <application-server>

Required? Optional; only one allowed

Child elements:

Specifies the transaction manager configuration file.

Table B–19 <transaction-manager-config> Attributes

Name	Description
path	Values: string Default: `transaction-manager.xml` The filename and path of the transaction manager configuration file. The default file is `ORACLE_HOME/j2ee/home/config/transaction-manager.xml`.

<web-site>

Parent element: <application-server>

Required? Optional; multiple allowed

Child elements:

References the configuration file for a single Web site defined within OC4J. A <web-site> element must be created for each Web site; otherwise, the site will not be enabled within OC4J. See Chapter 13, "Managing Web Sites in OC4J" for details.

Table B–20 <web-site> Attributes

Name	Description
path	Values: string Default: n/a The filename and path of the `*-web-site.xml` configuration file defining the Web site.

Overview of the Web Site Configuration Files (*-web-site.xml)

The element descriptions in this section apply to any OC4J Web site configuration file, including `default-web-site.xml`.

<web-site>

Required? Required; one only

Child elements:

```
<description>
<frontend>
<web-app>
<default-web-app>
<user-web-apps>
<access-log>
<odl-access-log>
<ssl-config>
```

This is the root element for a Web site configuration file.

Table B–21 Web Site Configuration File Attributes

Name	Description
`display-name`	Values: string Default: n/a Optionally defines a user-friendly or informal Web site name.
`host`	Values: string Default: n/a Specifies the host for this Web site, as either a DNS host name or an IP address. If a server is a "multi-home" machine (having multiple IP addresses), you can use the "`[ALL]`" setting to listen to all IP addresses.
`log-request-info`	Values: Boolean Default: `false` Specifies whether to write information about the incoming request into the Web site log if an error occurs. The Web site log is enabled through either the `<access-log>` or `<odl-access-log>` element, described later in this section. ("Enabling/Disabling Access Logging for a Web Module/Application" on page 13-15 provides additional information about enabling the Web site log.)

Table B–21 (Cont.) Web Site Configuration File Attributes

Name	Description
`max-request-size`	Values: string Default: `15000` Sets a maximum size, in bytes, for incoming HTTP requests. If a client sends a request that exceeds this maximum, it will receive a "request entity too large" error. The default maximum is 15000.
`secure`	Values: Boolean Default: `false` Specifies whether to support Secure Socket Layer (SSL) functionality. For a protocol setting of "`ajp13`" (used in an Oracle Application Server environment), a "`true`" setting results in secure AJP protocol between Oracle HTTP Server and OC4J. For a protocol setting of "`http`" (used in OC4J standalone), a "`true`" setting results in HTTPS protocol between the client and OC4J. Also note that a `secure="true"` setting requires that you use the `<ssl-config>` element (a subelement under the `<web-site>` element) to specify the keystore path and password. This element is documented later in this section. SSL and HTTPS features are also available through Oracle HTTP Server for communication between Oracle HTTP Server and the client. For information, see *Oracle Application Server Security Guide*.
`protocol`	Values: string Default: n/a Specifies the protocol that the Web site is using. Possible values are `"http"` and `"ajp13"` (for AJP, the default). In a production environment with Oracle Application Server, you should use only the "`ajp13`" setting. The AJP protocol is for use with Oracle HTTP Server and `mod_oc4j`. Note that each protocol must have a corresponding port, and each port must have a corresponding protocol. The "`http`" setting is for OC4J standalone. To use either an "`ajp13`" or "`http`" setting in secure mode (SSL), you must set the `secure` flag to "`true`" and use the `<ssl-config>` subelement to specify the keystore path and password. This element is documented later in this section.
`port`	Values: string Default: n/a Specifies the port number for this Web site. Each port must have a corresponding protocol, and each protocol must have a corresponding port. In OC4J standalone, a port setting of 8888 is used by default for direct access to the OC4J listener, but you can change this as desired. In an Oracle Application Server environment, this port setting is overridden by OPMN, the Oracle Process Management and Notification system. Oracle Application Server uses port `7777` by default for access through Oracle HTTP Server with Oracle Application Server Web Cache enabled. In a UNIX environment, port numbers less than `1024` require root privileges for access. Also note that if there is no port specification from the client browser, port `80` is assumed for HTTP protocol and port `443` for HTTPS.

Table B–21 (Cont.) Web Site Configuration File Attributes

Name	Description
`use-keep-alive`	Values: Boolean Default: `true` Typical behavior for a servlet container is to close a connection once a request has been completed. With a `use-keep-alive` setting of `"true"`, however, a connection is maintained across requests. For AJP protocol, connections are always maintained and this attribute is ignored. For other protocols, the default is `"true"`; disabling it may cause significant performance loss.
`virtual-hosts`	Values: string Default: `n/a` This optional attribute is useful for virtual sites sharing the same IP address. The value is a comma-delimited list of host names tied to this Web site.

<description>

Contains an optional brief description of the Web site.

<frontend>

Specifies a perceived front-end host and port of this Web site as seen by HTTP clients. When the site is behind a load balancer or firewall, the `<frontend>` specification is necessary to provide appropriate information to Web application code for functionality such as URL rewriting.

Using the host and port specified in the `<frontend>` element, the back-end server running the application knows to refer to the front end, instead of to itself, in any URL rewriting. This way, subsequent requests properly come in through the front end again, instead of trying to access the back end directly.

Table B–22 describes the attributes of `<frontend>`.

Table B–22 <frontend> Attributes

Name	Description
`host`	Values: string Default: `n/a` Specifies the host name of the front-end server, such as `"www.acme.com"`.
`port`	Specifies the port number of the front-end server, such as `"80"`.

<web-app>

This element binds a particular Web module to this Web site. It specifies the name of a J2EE application archive (EAR file name minus the `.ear` extension) from the `server.xml` file, and the name of a Web module within the J2EE application. The Web module is defined in the J2EE `application.xml` file in the application EAR file (or possibly in the `orion-application.xml` file in the EAR file). The Web module is bound at the location specified by the `<web-app>` element `root` attribute.

Note: It is possible to deploy a WAR file by itself, instead of within an EAR file. In OC4J standalone, such Web applications are added to the OC4J default application. (In OC4J, there must always be a parent application of some sort.) See "Understanding the Application Hierarchy in OC4J" on page 1-8 for more information.

In this scenario, the Web site XML file `<web-app>` element specifies the name of the default application rather than the name of a J2EE application archive. More details are provided in the attribute descriptions and examples that follow.

Mapping to and from Web site XML files, particularly with respect to the `application` and `name` attributes, is shown in examples elsewhere in this document. See "Deploying a J2EE Application (EAR)" on page 6-5 (for a typical scenario of deploying a WAR file within an EAR file) and "Deploying a Standalone Web Module (WAR)" on page 6-7 (for the scenario of deploying a WAR file by itself to the OC4J default application).

Table B–23 describes the attributes of `<web-app>`.

Table B–23 <web-app> Attributes

Name	Description
access-log	Values: string Default: true Specifies whether OC4J access logging, which logs requests to the Web site, is enabled for the Web module. The default is `true`. If log file management becomes an issue, set to `false` to disable access logging for the module. See the descriptions of the `<access-log>` and `<odl-access-log>` elements within this section for more on access log configuration.
application	Values: string Default: n/a Specifies the J2EE application archive name, which is the EAR file name without the `.ear` extension, and which corresponds to the `name` attribute of an `<application>` element in the `server.xml` file. If you deploy a WAR file by itself in OC4J standalone, using the OC4J default application as the parent, then the `application` attribute instead reflects the name of the default application, according to the `<global-application>` element in the `server.xml` file.
load-on-startup	Values: Boolean Default: `false` Optional. Specifies whether the Web module should be preloaded on application startup. Otherwise, it is loaded upon the first request for it. Supported values are `"true"` and `"false"`. The default is `false`; however, this value is explicitly set to `true` when the module/application is deployed through Oracle Enterprise Manager 10*g* Application Server Control Console.

Table B–23 (Cont.) <web-app> Attributes

Name	Description
max-inactivity-time	Values: string Default: `0` Optional. Specifies the number of minutes of inactivity after which OC4J will shut down the Web module. By default, a Web module is never shut down due to inactivity.
name	Values: Boolean Default: n/a Specifies the name of a Web module within the specified J2EE application, and corresponds to the `<web-uri>` value (without the `.war` extension) of a `<web>` subelement of a `<module>` element in the J2EE `application.xml` file. The J2EE `application.xml` file is in the EAR file.
`root`	Values: string Default: n/a Specifies the path to which the Web module is to be bound, which defines the context path portion of the URL used to invoke the module. For example, if the Web module `CatalogApp` at Web site `www.example.com` is bound to the `root` setting `"/catalog"`, then it can be invoked as follows: `http://www.example.com/catalog` The `root` attribute overrides the `<context-root>` value of the corresponding `<web>` element in the J2EE `application.xml` file. Even though the `<context-root>` element is mandatory in an `application.xml` file, its value is not used by OC4J. Specifying a `root` setting of `"/"` will override the OC4J default Web application. This setting or a null setting is not allowed by the `admin.jar` utility when binding a Web application to the Web site.
shared	Values: string Default: `false` Allows sharing of a published Web module between Web sites, when a Web site is defined by a particular pairing of a protocol and a port. Supported values are `"true"` and `"false"` (default). Use `shared="true"` only in OC4J standalone. If an HTTPS Web application is marked as shared, its session tracking strategy reverts from SSL session tracking to session tracking through cookies or URL rewriting. This could possibly make the Web application less secure but may be necessary to work around issues such as SSL session timeouts not being properly supported in some browsers.

<default-web-app>

This element creates a reference to the default Web application bound to this Web site. When a single application is bound to the Web site - such as Application Server Control Console - specify the application within this element.

For users, this element is relevant only in an OC4J standalone environment. In an Oracle Application Server environment, the OC4J default Web application has system-level functionality but is not otherwise meaningful.

The <default-web-app> element uses the same attributes as the <web-app> element described immediately preceding, but note that the default setting of load-on-startup is "true".

<user-web-apps>

Use this element to support user directories and applications. Each user has his or her own Web module and associated web-application.xml file. User applications are reached at /*username*/ from the server root.

Table B–24 describes the attributes of <user-web-apps>.

Table B–24 <user-web-apps> Attributes

Name	Description
max-inactivity-time	Values: int Default: n/a Optional integer attribute to specify the number of minutes of inactivity after which OC4J will shut down the Web module. By default, a Web module is never shut down due to inactivity.
path	Specifies a path to specify the local directory of the user application, including a wildcard for the user name. The default path setting in UNIX, for example, is */home/username*, where *username* is replaced by the particular user name.

<access-log>

Use this element to enable text-based access logging for this Web site and to specify information about the access log, including the path, file name, and what information is included. The log file is where incoming requests (each access of the Web site) are logged.

See "Configuring Text-Based Access Logging" on page 13-13 for configuration details.

<odl-access-log>

Use this element to enable ODL-based access logging for the Web site and to specify information about the access logs, including the path, and maximum values for the size of each file and the total size of all files in the log directory. The log files are where incoming requests (each access of the Web site) are logged.

See "Configuring ODL Access Logging" on page 13-14 for configuration details.

<ssl-config>

This element specifies SSL configuration settings, if applicable. You must use it whenever you set the secure attribute of the <web-site> element to "true".

Subelement of <ssl-config>:

<property>

Table B–25 describes the attributes of <ssl-config>.

Table B–25 <ssl-config> Attributes

Name	Description
keystore	Values: string Default: n/a A relative or absolute path to the keystore database (a binary file) used by this Web site to store certificates and keys for the user base in this installation. The path value includes the file name. A relative path is relative to the location of the Web site XML file. A keystore is a `java.security.KeyStore` instance and can be created and maintained using the `keytool` utility, provided with the Sun Microsystems JDK
`keystore-password`	Values: string Default: n/a The password required to open the keystore.
`needs-client-auth`	Values: string Default: `false` Indicates whether the entity that is a client to OC4J, such as Oracle HTTP Server, must submit a certificate for authorization so it can communicate with OC4J. Supported values are `"true"` for "client authentication" (certificate required) and `"false"`, the default (no certificate required).
`provider`	Values: string Default: `com.sun.net.ssl.internal.ssl.Provider` You can use this attribute to specify a provider if you are using JSSE (Java Secure Socket Extension). By default, OC4J usually employs the Sun Microsystems implementation of SSL. However, OC4J employs the Oracle SSL implementation in some cases, such as for SOAP and `http_client`.
`factory`	Values: string Default: `JSSE: com.evermind.ssl.JSSESSLServerSocketFactory` If you are not using JSSE, use the `factory` attribute to specify an implementation of `SSLServerSocketFactory`. If you use a third-party `SSLServerSocketFactory` implementation, you can use `<property>` subelements of the `<ssl-config>` element to send parameters to the factory.

C

Overview of the Session State Tables

This appendix documents the schema for the database tables used by the OC4J database persistence mechanism. See "Configuring Database Replication" on page 9-9 for additional information on this mechanism.

OC4J_HTTP_SESSION

This table stores metadata for a single HTTP session, including identifiers for the application and user setting properties on the session. The `ID` is the primary key.

There is a 1:many relationship between an `OC4J_HTTP_SESSION` table and the `OC4J_HTTP_SESSION_VALUE` tables. Each entry in the `OC4J_HTTP_SESSION` table may have 0 or more entries in the `OC4J_HTTP_SESSION_VALUE` table.

Table C–1 OC4J_HTTP_SESSION Table Description

Name	Null?	Data Type	Description
`ID`	`NOT_NULL`	`VARCHAR2(100)`	The unique session ID.
`APPLICATION_ID`	`NULL`	`VARCHAR2(100)`	The OC4J internal ID assigned to the application the session belongs to.
`IP`	`NULL`	`NUMBER(38)`	The IP address of the machine hosting the application.
`LAST_ACCESSED`	`NULL`	`NUMBER(38)`	The last time the current record was updated.
`USER_NAME`	`NULL`	`VARCHAR2(50)`	The user name for the application user setting values on the session.
`MAX_INACTIVE_TIME`	`NULL`	`NUMBER(38)`	The maximum time the session can remain idle before being expired. Session data will not be persisted after this maximum is exceeded.
`CREATION_TIME`	`NULL`	`NUMBER(38)`	The time at which the table was created.

OC4J_HTTP_SESSION_VALUE

This table stores each HTTP session property and the values set on it by the application user. The values are stored as a BLOB (Binary Large OBject). The `ID` and `KEY_FIELD` together comprise the primary key.

Table C–2 OC4J_HTTP_SESSION_VALUETable Description

Name	Null?	Data Type	Description
ID	NOT_NULL	VARCHAR2(100)	The unique session ID.
KEY_FIELD	NOT_NULL	VARCHAR2(100)	The name of a property set by the application user on the session.
VALUE_FIELD	NULL	BLOB	The value of the property set on the session.

OC4J_EJB_SESSION

This table stores the current state of a stateful session bean. The state data is stored as a BLOB (Binary Large OBject). The ID is the primary key.

Table C–3 OC4J_EJB_SESSION Table Description

Name	Null?	Data Type	Description
ID	NOT_NULL	VARCHAR2(100)	The unique session ID.
VALUE_FIELD	NULL	BLOB	The current state data of the session bean.
LOCATION	NULL	NUMBER(38)	The JNDI name that the session bean is bound to.
CHECKSUM	NULL	NUMBER(38)	Used internally to validate that bytes are formatted correctly.
PASSIVATE	NULL	NUMBER(38)	A Boolean value indicating whether the bean has been passivated. If true, the passivated bean will be retrieved from disk.
LAST_ACCESSED	NULL	NUMBER(38)	The last time the current record was updated.
USER_NAME	NULL	VARCHAR2(50)	The user name for the application user setting values on the session.
MAX_INACTIVE_TIME	NULL	NUMBER(38)	The maximum time the session can remain idle before being expired. Session data will not be persisted after this maximum is exceeded.
CREATION_TIME	NULL	NUMBER(38)	The time at which the table was created.

D

Third Party Licenses

This appendix includes the Third Party License for all the third party products included with Oracle Application Server.

ANTLR

This program contains third-party code from ANTLR. Under the terms of the Apache license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the ANTLR software, and the terms contained in the following notices do not change those rights.

The ANTLR License

```
Software License

We reserve no legal rights to the ANTLR--it is fully in the public domain. An
individual or company may do whatever they wish with source code distributed with
ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its
output, into commerical software.
We encourage users to develop software with ANTLR. However, we do ask that credit
is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or
incorporate any source code into one of your programs (commercial product,
research project, or otherwise) that you acknowledge this fact somewhere in the
documentation, research report, etc... If you like ANTLR and have developed a nice
tool with the output, please mention that you developed it using ANTLR. In
addition, we ask that the headers remain intact in our source code. As long as
these guidelines are kept, we expect to continue enhancing this system and expect
to make other tools available as they are completed.
```

Apache

This program contains third-party code from the Apache Software Foundation ("Apache"). Under the terms of the Apache license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache software, and the terms contained in the following notices do not change those rights.

The Apache license agreements apply to the following included Apache components:

- Apache HTTP Server
- Apache JServ
- mod_jserv

- Regular Expression package version 1.3
- Apache Expression Language packaged in commons-el.jar
- mod_mm 1.1.3
- Apache XML Signature and Apache XML Encryption v. 1.4 for Java and 1.0 for C++
- log4j 1.1.1
- BCEL v. 5
- XML-RPC v. 1.1
- Batik v. 1.5.1
- ANT 1.6.2 and 1.6.5
- Crimson v. 1.1.3
- ant.jar
- wsif.jar
- bcel.jar
- soap.jar
- Jakarta CLI 1.0
- jakarta-regexp-1.3.jar
- JSP Standard Tag Library 1.0.6 and 1.1
- Struts 1.1
- Velocity 1.3
- svnClientAdapter
- commons-logging.jar
- wsif.jar
- commons-el.jar
- standard.jar
- jstl.jar

The Apache Software License

License for Apache Web Server 1.3.29

```
/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000-2002 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
```

```
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 *    nor may "Apache" appear in their name, without prior written
 *    permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * ====================================================================
 *
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation.  For more
 * information on the Apache Software Foundation, please see
 * <http://www.apache.org/>.
 *
 * Portions of this software are based upon public domain software
 * originally written at the National Center for Supercomputing
Applications,
 * University of Illinois, Urbana-Champaign.
```

License for Apache Web Server 2.0

```
Copyright (c) 1999-2004, The Apache Software Foundation
Licensed under the Apache License, Version 2.0 (the "License"); you may not use
this file except in compliance with the License.  You may obtain a copy of the
License at http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed
under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied.  See the License for the
specific language governing permissions and limitations under the License.
Copyright (c) 1999-2004, The Apache Software Foundation
                                 Apache License
                            Version 2.0, January 2004
                         http://www.apache.org/licenses/
```

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Apache SOAP

This program contains third-party code from the Apache Software Foundation ("Apache"). Under the terms of the Apache license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Apache

software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Apache.

Apache SOAP License

Apache SOAP license 2.3.1

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

DBI Module

This program contains third-party code from Tim Bunce. Under the terms of the Tim Bunce license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Tim Bunce software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Tim Bunce software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Tim Bunce.

The DBI module is Copyright (c) 1994-2002 Tim Bunce. Ireland. All rights reserved.

You may distribute under the terms of either the GNU General Public License or the Artistic License, as specified in the Perl README file.

Perl Artistic License

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b. use the modified Package only within your corporation or organization.

c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d. make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b. accompany the distribution with the machine-readable source of the Package with your modifications.

c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d. make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package through the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt

is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

FastCGI

This program contains third-party code from Open Market, Inc. Under the terms of the Open Market license, Oracle is required to license the Open Market software to you under the following terms. Note that the terms contained in the Oracle program license that accompanied this product do not apply to the Open Market software, and your rights to use the software are solely as set forth below. Oracle is not responsible for the performance of the Open Market software, does not provide technical support for the software, and shall not be liable for any damages arising out of any use of the software.

FastCGI Developer's Kit License

This FastCGI application library source and object code (the "Software") and its documentation (the "Documentation") are copyrighted by Open Market, Inc ("Open Market"). The following terms apply to all files associated with the Software and Documentation unless explicitly disclaimed in individual files.

Open Market permits you to use, copy, modify, distribute, and license this Software and the Documentation solely for the purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions.

No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this Software and Documentation may be copyrighted by their authors and need not follow the licensing terms described here, but the modified Software and Documentation must be used for the sole purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose. If modifications to this Software and Documentation have new licensing terms, the new terms must protect Open Market's proprietary rights in the Software and Documentation to the same extent as these licensing terms and must be clearly indicated on the first page of each file where they apply.

Open Market shall retain all right, title and interest in and to the Software and Documentation, including without limitation all patent, copyright, trade secret and other proprietary rights.

OPEN MARKET MAKES NO EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE SOFTWARE OR THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL OPEN MARKET BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DAMAGES ARISING FROM OR RELATING

TO THIS SOFTWARE OR THE DOCUMENTATION, INCLUDING, WITHOUT LIMITATION, ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR SIMILAR DAMAGES, INCLUDING LOST PROFITS OR LOST DATA, EVEN IF OPEN MARKET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS". OPEN MARKET HAS NO LIABILITY IN CONTRACT, TORT, NEGLIGENCE OR OTHERWISE ARISING OUT OF THIS SOFTWARE OR THE DOCUMENTATION.

Module mod_fastcgi License

This FastCGI application library source and object code (the "Software") and its documentation (the "Documentation") are copyrighted by Open Market, Inc ("Open Market"). The following terms apply to all files associated with the Software and Documentation unless explicitly disclaimed in individual files.

Open Market permits you to use, copy, modify, distribute, and license this Software and the Documentation solely for the purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions.

No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this Software and Documentation may be copyrighted by their authors and need not follow the licensing terms described here, but the modified Software and Documentation must be used for the sole purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose. If modifications to this Software and Documentation have new licensing terms, the new terms must protect Open Market's proprietary rights in the Software and Documentation to the same extent as these licensing terms and must be clearly indicated on the first page of each file where they apply.

Open Market shall retain all right, title and interest in and to the Software and Documentation, including without limitation all patent, copyright, trade secret and other proprietary rights.

OPEN MARKET MAKES NO EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE SOFTWARE OR THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL OPEN MARKET BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DAMAGES ARISING FROM OR RELATING TO THIS SOFTWARE OR THE DOCUMENTATION, INCLUDING, WITHOUT LIMITATION, ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR SIMILAR DAMAGES, INCLUDING LOST PROFITS OR LOST DATA, EVEN IF OPEN MARKET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS". OPEN MARKET HAS NO LIABILITY IN CONTRACT, TORT, NEGLIGENCE OR OTHERWISE ARISING OUT OF THIS SOFTWARE OR THE DOCUMENTATION.

Info-ZIP Unzip Package

This program contains third-party code from Info-ZIP. Under the terms of the Info-ZIP license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Info-ZIP software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the

contrary in the Oracle program license, the Info-ZIP software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Info-ZIP.

The Info-ZIP Unzip Package License

```
Copyright (c) 1990-1999 Info-ZIP. All rights reserved. For the purposes of this
copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup
Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig,
Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno
van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens,
George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith,
Christian Spieler, Antoince Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "AS IS," without warranty of any kind, express or
implied.  In no event shall InfoZIP or its contributors be held liable for any
direct, indirect, incidental, special or consequential damages arising out of the
use of or inability to use this software."
```

JSR 110

This program contains third-party code from IBM Corporation ("IBM"). Under the terms of the IBM license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the IBM software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the IBM software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or IBM.

```
Copyright IBM Corporation 2003 - All rights reserved

Java APIs for the WSDL specification are available at:
http://www-124.ibm.com/developerworks/projects/wsdl4j/
```

Jaxen

This program contains third-party code from the Apache Software Foundation ("Apache") and from the Jaxen Project ("Jaxen"). Under the terms of the Apache and Jaxen licenses, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache and Jaxen software, and the terms contained in the following notices do not change those rights.

The Jaxen License

```
Copyright (C) 2000-2002 bob mcwhirter & James Strachan. All rights reserved.
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list
of conditions, and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this
list of conditions, and the disclaimer that follows these conditions in the
documentation and/or other materials provided with the distribution.

The name "Jaxen" must not be used to endorse or promote products derived from this
```

software without prior written permission. For written permission, please contact license@jaxen.org.

Products derived from this software may not be called "Jaxen", nor may "Jaxen" appear in their name, without prior written permission from the Jaxen Project Management (pm@jaxen.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgment equivalent to the following: "This product includes software developed by the Jaxen Project (http://www.jaxen.org/)." Alternatively, the acknowledgment may be graphical using the logos available at http://www.jaxen.org/.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE Jaxen AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Jaxen Project and was originally created by bob mcwhirter and James Strachan . For more information on the Jaxen Project, please see http://www.jaxen.org/.

JGroups

This program contains third-party code from GNU. Under the terms of the GNU license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the GNU software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the GNU software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or GNU.

The GNU License

GNU Lesser General Public License
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages

in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a

program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the

Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept

this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask

for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Libraries
If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and an idea of what it does.> Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

mod_mm and mod_ssl

This program contains third-party code from Ralf S. Engelschall ("Engelschall"). Under the terms of the Engelschall license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product

determines your right to use the Oracle program, including the Engelschall software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the mod_mm software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Engelschall.

mod_mm

```
Copyright (c) 1999 - 2000 Ralf S. Engelschall.  All rights reserved.
This product includes software developed by Ralf S. Engelschall
<rse@engelschall.com> for use in the mod_ssl project (http://www.modssl.org/).
```

mod_ssl

```
Copyright (c) 1998-2001 Ralf S. Engelschall.  All rights reserved.
This product includes software developed by Ralf S. Engelschall
<rse@engelschall.com> for use in the mod_ssl project (http://www.modssl.org/).
```

OpenSSL

This program contains third-party code from the OpenSSL Project. Under the terms of the OpenSSL Project license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the OpenSSL software, and the terms contained in the following notices do not change those rights.

OpenSSL License

```
/* ====================================================================
 * Copyright (c) 1998-2005 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
```

```
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

 Original SSLeay License
 -----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
```

```
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

Perl

This program contains third-party code from the Comprehensive Perl Archive Network ("CPAN"). Under the terms of the CPAN license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the CPAN software, and the terms contained in the following notices do not change those rights.

Perl Kit Readme

Copyright 1989-2001, Larry Wall

All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of either:

1. the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
2. the "Artistic License" which comes with this Kit.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See either the GNU General Public License or the Artistic License for more details.

You should have received a copy of the Artistic License with this Kit, in the file named "Artistic". If not, I'll be glad to provide one.

You should also have received a copy of the GNU General Public License along with this program in the file named "Copying". If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA or visit their Web page on the internet at http://www.gnu.org/copyleft/gpl.html.

For those of you that choose to use the GNU General Public License, my interpretation of the GNU General Public License is that no Perl script falls under the terms of the GPL unless you explicitly put said script under the terms of the GPL yourself. Furthermore, any object code linked with perl does not automatically fall under the terms of the GPL, provided such object code only adds definitions of subroutines and variables, and does not otherwise impair the resulting interpreter from executing any standard Perl script. I consider linking in C subroutines in this manner to be the moral equivalent of defining subroutines in the Perl language itself. You may sell such an object file as proprietary provided that you provide or offer to provide the Perl source, as specified by the GNU General Public License. (This is merely an alternate way of specifying input to the program.) You may also sell a binary produced by the dumping of a running Perl script that belongs to you, provided that you provide or offer to provide the Perl source as specified by the GPL. (The fact that a Perl interpreter and your code are in the same binary file is, in this case, a form of mere aggregation.) This is my interpretation of the GPL. If you still have concerns or difficulties understanding my intent, feel free to contact me. Of course, the Artistic License spells all this out for your protection, so you may prefer to use that.

mod_perl 1.29 License

```
/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 1996-2000 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 *    nor may "Apache" appear in their name, without prior written
 *    permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
```

```
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * ====================================================================
 */
```

mod_perl 1.99_16 License

```
Copyright (c) 1999-2004, The Apache Software Foundation
Licensed under the Apache License, Version 2.0 (the "License"); you may not use
this file except in compliance with the License.  You may obtain a copy of the
License at http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed
under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied.  See the License for the
specific language governing permissions and limitations under the License.
Copyright (c) 1999-2004, The Apache Software Foundation
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).
```

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

```
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.
```

Perl Artistic License

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b. use the modified Package only within your corporation or organization.

c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d. make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b. accompany the distribution with the machine-readable source of the Package with your modifications.

c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d. make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package through the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt

is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

SAXPath

This program contains third-party code from SAXPath. Under the terms of the SAXPath license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the SAXPath software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the SAXPath software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or SAXPath.

The SAXPath License

```
Copyright (C) 2000-2002 werken digital. All rights reserved. Redistribution and
use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list
of conditions, and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this
list of conditions, and the disclaimer that follows these conditions in the
documentation and/or other materials provided with the distribution.

The name "SAXPath" must not be used to endorse or promote products derived from
this software without prior written permission. For written permission, please
contact license@saxpath.org.

Products derived from this software may not be called "SAXPath", nor may "SAXPath"
appear in their name, without prior written permission from the SAXPath Project
Management (pm@saxpath.org).

In addition, we request (but do not require) that you include in the end-user
documentation provided with the redistribution and/or in the software itself an
acknowledgment equivalent to the following: "This product includes software
developed by the SAXPath Project (http://www.saxpath.org/)." Alternatively, the
acknowledgment may be graphical using the logos available at
http://www.saxpath.org/.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE SAXPath
AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
```

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the SAXPath Project and was originally created by bob mcwhirter and James Strachan . For more information on the SAXPath Project, please see http://www.saxpath.org/.

W3C DOM

This program contains third-party code from the World Wide Web Consortium ("W3C"). Under the terms of the W3C license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the W3C software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the W3C software is provided by Oracle AS IS and without warranty or support of any kind from Oracle or W3C.

The W3C License

W3C® SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231
This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)
THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Index

Symbols

<cluster> element, 9-10
<file> element, 10-3
<global-thread-pool> element, 11-2, 11-4
<log> element, 10-3, 10-4
<odl> element, 10-4
<session-tracking> element, 13-9
<web-app> element, 13-8
<work-manager-thread-pool> element, 11-4

A

access logging, 13-12
  configuring, 13-12
  disabling for a Web module, 13-15
  ODL format, 13-14
  text-based, 13-13
admin_client.jar tool
  using, 3-3, 6-1
admin.jar tool
  administration, 3-4
  shut down, 5-2
  -site option, 7-8
  undeploying with, 6-11
  using, 7-1
AJP
  overview, 1-8
Apache JServ protocol, see AJP
application mount points, 8-26
Application Server Control Console
  creating a Web site for, 13-7
  overview, 3-1
  starting and stopping, 6-14
applications
  binding to multiple Web sites, 13-8
  deploying, 6-5, 7-3
  restarting, 7-6
  starting and stopping, 6-14
  system application, B-2
  undeploying, 6-5, 7-3
associateUsingThirdTable property, 4-6

C

clustering
  changes in 10.1.3, 8-4
  database replication, 9-9
  database schema, C-1
  disabling, 9-10
  dynamic node discovery, 8-5
  dynamic peer-to-peer replication, 9-7
  islands, 9-1
  JGroups, 9-6
  multicast replication, 9-5
  node configuration, 8-1
  overview, 9-1
  peer-to-peer replication, 9-6
  replication options, 9-3
  topologies, 8-1
command-line options, 4-4
configuration files
  changes in the current release, 1-4
  list of OC4J-specific, B-3
  overview of, B-1
  server.xml, B-4
  system-application.xml, 1-9
connection thread pool, 11-2
cookie domain, 13-9
cookie-domain attribute, 13-9
createinstance utility, 8-35
creating additional OC4J instances, 8-35

D

data sources
  converting to new configuration, 7-12
  creating for an application, 7-10
  removing, 7-11
  testing, 7-11
dedicated.connection setting, 4-5
dedicated.rmicontext property, 4-5
default application, 1-9
default-web-site.xml file, 13-1
DefineColumnType property, 4-6
DTDs
  registering with OC4J, 14-1

E

EJB 3.0 support, 1-3
EJB module

updating, 6-11, 7-7
entity resolver, 14-1
environment variables
J2EE_HOME, 2-2
JAVA_HOME, 2-1
ORACLE_HOME, 2-1
setting, 2-1

G

garbage collection
impact on server performance, A-2
GenerateIIOP property, 4-5
global web application, 1-9

H

HTTP method
trace, 4-8
http.method.trace.allow property, 4-8
http.request.debug property, 4-7, 4-8
HTTPS, 13-9
client-authentication, 13-11

I

InitialContext, 4-5
instances
creating additional OC4J instances, 8-35

J

J2EE
definition, 1-1
supported APIs, 1-2
J2EE Management support, 12-1
J2EE_HOME environment variable, 2-2
j2ee-logging.xml, 10-8
Java Management Extensions support, 12-1
JAVA_HOME environment variable, 2-1
java.ext.dirs property, 4-5
java.io.tmpdir property, 4-5
java.lang.OutOfMemory errors, A-1
JDK
supported versions, 1-1
JMX notifications
subscribing to, 12-5
JMX support, 12-1
JSR-77 support, 12-1
JVM, 1-1

K

KeepIIOPCode property, 4-5

L

load balancing, 8-21
loadbalancer.jar, 9-2
logging
Oracle Diagnostic Logging, 10-4
plain text, 10-3
rotating log files, 10-4
summary of log files, 10-1

M

MBeans
accessing, 12-5
application-specific, 12-6
notifications, 12-5
persistence policy, 12-4
using, 12-5
what are, 12-1
mod_oc4j module, 1-8, 8-21
mount points, 8-26

N

needs-client-auth attribute, 13-11
Network Interface Card binding, 9-6

O

OC4J
administration, 3-3, 3-4, 6-1, 7-1
command-line interface, 3-3, 6-1, 7-1
command-line options, 4-4
creating new instances of, 8-35
restarting, 5-3
shutting down, 5-2
startup, 5-1
system properties, 4-4
troubleshooting, A-1
OC4J executable scripts, 3-4
OC4J version, 7-3
OC4J_EJB_SESSION, C-2
oc4j_extended, 2-1
OC4J_HTTP_SESSION, C-1
OC4J_HTTP_SESSION_VALUE, C-1
oc4j.cmd batch file
using, 3-4
oc4j-connectors.xml, B-3
oc4j.jar tool
startup, 5-1
oc4j-ra.xml, B-4
oc4j.sh shell script
using, 3-4
OHS
dynamic OC4J discovery, 8-26
load balancing, 8-21
mod_oc4j module, 8-21
overview, 1-8
routing IDs, 8-22
ONS, 8-2
opmnassociate tool, 8-8
opmnctl
config port command, 13-4
configuring a cluster, 8-9
configuring Web site ports, 13-4
using to start OC4J, 5-2
OPMN-managed replication, 9-7

Oracle Diagnostic Logging, 10-4
Oracle Diagnostic Logging, see logging
  ODL
Oracle HTTP Server, 1-8
oracle Logger
  configuring, 10-8
Oracle Notification System, 8-2
ORACLE_HOME environment variable, 2-1
oracle.dms.gate setting, 4-6
oracle.dms.sensors setting, 4-6
oracle.mdb.fastUndeploy property, 4-5
oracle-webservices.xml, B-4
Out of Memory error, 4-5

P

performance
  oracle.dms.sensors setting, 4-6
performance setting
  dedicated.connection, 4-5
  dedicated.rmicontext, 4-5
  DefineColumnType, 4-6
  oracle.dms.gate, 4-6
  task manager granularity, 11-1
  thread pools, 11-2

R

removeinstance utility, 8-39
resource adapters
  deploying, 7-12
  undeploying, 7-12
restarting
  OC4J, 5-3
routing IDs
  creating, 8-22
  using to manage Web routing, 8-22

S

schemas
  viewing, B-1
Secure Socket Layer--see SSL
secure Web site, 13-9
security
  OC4J and OHS configuration, 13-10
server.xml file, B-4
SSL, 13-9
  client-authentication, 13-11
starting OC4J in a standalone environment, 5-1
stopping OC4J in a standalone configuration, 5-2
system application, 1-8, B-2
  configuring, 1-8
  overview, 1-8
System MBean Browser, 12-5
system properties, 4-4
system-application.xml, 1-9, B-2

T

task manager
  execution interval, 11-1
  setting granularity, 11-1
taskmanager-granularity attribute, 11-1
thread pooling
  configuring, 11-2
troubleshooting OC4J, A-1
two-phase commit transaction coordinator, 1-4

V

version
  view OC4J release version, 7-3

W

Web communication
  in a standalone OC4J installation, 1-6
  in an Oracle Application Server installation, 1-8
Web server
  native OC4J listener, 1-6
  Oracle HTTP Server, 1-8
Web services support, 1-3
Web site
  binding applications to multiple sites, 13-8
  client authentication, 13-11
  configuration file, 13-6
  configuring in OPMN, 13-2
  configuring in standalone OC4J, 13-2
  creating a new site, 13-5
  referencing in opmn.xml, 13-7
  referencing in server.xml, 13-7
  secure, 13-9
  starting and stopping, 13-12
work management thread pool, 11-4
worker thread pool, 11-2
ws.debug property, 4-8

X

XML files
  reloading modified, 7-3
XML schemas, B-1
XSDs
  registering with OC4J, 14-1
  viewing, B-1