

**Oracle® Containers for J2EE**

Deployment Guide

10g Release 3 (10.1.3)

**Part No. B14431-01**

January 2006

Oracle Containers for J2EE Deployment Guide, 10g Release 3 (10.1.3)

Part No. B14431-01

Copyright © 2006, Oracle. All rights reserved.

Primary Author: Dan Hynes

Contributors: Steve Button, Tugdual Grall, Lars Ewe, Gerald Ingalls, Mike Lehmann, Jianmin Liu, Angela Long, Jasen Minton, Debu Panda, Shiva Prasad, Chaya Ramanujam, Merrick Schincariol, Charlie Shapiro, Gael Stevens, Sindhu Subramanyam

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

---

---

# Contents

<b>Preface</b> .....	ix
Intended Audience.....	ix
Documentation Accessibility .....	ix
Related Documents .....	x
Conventions .....	xi
<b>1 Getting Started</b>	
<b>Understanding Deployment in OC4J</b> .....	1-1
Valid Components for Deployment .....	1-1
Support for the J2EE Application Deployment API (JSR-88) in OC4J.....	1-1
“Hot Deployment” in OC4J .....	1-2
<b>Options for Deploying into OC4J</b> .....	1-2
Application Server Control Console User Interface.....	1-2
OC4J Ant Tasks.....	1-3
admin_client.jar Command Line Utility .....	1-3
admin.jar Command Line Utility.....	1-3
Oracle JDeveloper .....	1-4
<b>2 Deploying A J2EE Application Into OC4J</b>	
<b>Overview of the Application Deployment Process</b> .....	2-1
Designating a Parent Application.....	2-1
Binding the Web Application to a Web Site.....	2-2
Creating or Applying a Deployment Plan.....	2-2
Dynamic HTTP Server Mount Point Configuration in Oracle Application Server.....	2-2
<b>What Happens During Deployment?</b> .....	2-3
Example of a Deployed Application Directory Structure .....	2-4
<b>Redeploying an Application</b> .....	2-4
Impact of Redeploying a Parent Application.....	2-5
<b>Undeploying an Application</b> .....	2-5
Impact of Undeploying a Parent Application.....	2-5
<b>3 Deploying Large Applications</b>	
<b>Specifying the Compilation Mode to Use</b> .....	3-1
<b>Configuring the Java Compiler</b> .....	3-2
Specifying an Alternative Java Compiler .....	3-2

Configuring Out-of-process Versus In-process Compiler Execution.....	3-2
Summary of Java Compiler Configuration Parameters .....	3-3
<b>Tuning the OC4J JVM for Large Deployments .....</b>	<b>3-3</b>
<b>4 Deploying Enterprise JavaBeans</b>	
<b>Overview of EJB Deployment.....</b>	<b>4-1</b>
<b>Generating Client-side IIOP Stubs.....</b>	<b>4-2</b>
Generating Stubs with Application Server Control .....	4-2
Generating Stubs with admin_client.jar .....	4-2
Generating Stubs with the OC4J Ant Tasks .....	4-3
<b>Incremental Redeployment of Updated EJB Modules.....</b>	<b>4-3</b>
<b>Impact of EJB Redeployment on Application Clients .....</b>	<b>4-4</b>
Redeploying Session Beans.....	4-4
Stateless Session Beans .....	4-4
Stateful Session Beans .....	4-4
Redeploying Entity Beans .....	4-4
Bean-Managed Persistence Beans.....	4-5
Container-Managed Persistence Beans.....	4-5
<b>5 Deploying Web Modules</b>	
Deploying a Standalone Web Module .....	5-1
Redeploying a Standalone Web Application.....	5-1
Redeploying an Updated Web Module into an Existing Application .....	5-2
Adding or Modifying JavaServer Pages in an Active Web Module.....	5-2
Optionally Setting JSP Configuration Parameters in the XML Configuration File .....	5-3
<b>6 Deploying Resource Adapters</b>	
Deploying a Standalone RAR.....	6-1
Redeploying/Undeploying a Standalone RAR .....	6-1
<b>7 Deploying Web Services</b>	
Deploying a Web Service.....	7-1
When Should You Redeploy a Web Service?.....	7-1
<b>8 Working With Deployment Plans</b>	
<b>What Is a Deployment Plan? .....</b>	<b>8-1</b>
How Do Deployment Plans Interact with Packaged Deployment Descriptors? .....	8-2
Overview of J2EE and OC4J Deployment Descriptors.....	8-2
<b>Creating or Editing a Deployment Plan.....</b>	<b>8-3</b>
Creating or Editing Deployment Plans with Application Server Control Console .....	8-4
Creating or Editing Deployment Plans with Oracle JDeveloper .....	8-4
<b>Setting Properties in a Deployment Plan .....</b>	<b>8-4</b>
Setting J2EE Application Configuration Properties .....	8-5
Setting Web Module Configuration Properties .....	8-11
Setting Enterprise JavaBeans Module Configuration Properties .....	8-16

General EJB Properties .....	8-17
Setting Entity Bean Properties.....	8-18
Setting Session Bean Properties .....	8-22
Setting Message-Driven Bean Properties .....	8-24
Setting Web Services Configuration Properties.....	8-25
Setting General Web Services Properties .....	8-25
Setting Web Service Description Properties.....	8-26
Setting Application Client Configuration Properties .....	8-26
Setting Resource Adapter Properties .....	8-27

## 9 Deploying with Application Server Control Console

<b>Accessing Application Server Control Console .....</b>	<b>9-1</b>
Accessing Application Server Control Console in Standalone OC4J .....	9-2
Accessing Application Server Control Console in Oracle Application Server .....	9-2
<b>Deploying to OC4J Instances Within a Cluster.....</b>	<b>9-2</b>
<b>Deploying/Redeploying an Application .....</b>	<b>9-3</b>
<b>Completing the Deployment Tasks .....</b>	<b>9-4</b>
Selecting the Security Provider .....	9-4
Mapping Security Roles to Users and Groups.....	9-5
Configuring Enterprise JavaBeans Included in the Application.....	9-5
Configuring Class Loading/Shared Libraries .....	9-6
Configuring Clustering .....	9-7
Providing Resource Mappings.....	9-8
<b>Undeploying an Application .....</b>	<b>9-8</b>

## 10 Deploying with the OC4J Ant Tasks

<b>Preparing to Use the Ant Tasks.....</b>	<b>10-1</b>
Overview of the Ant Tasks .....	10-1
Prerequisites for Using the Ant Tasks.....	10-2
Incorporating the OC4J Ant Tasks into Your Environment .....	10-2
Incorporating the Ant Tasks Using Ant 1.6.2 Outside OC4J .....	10-2
Enabling Logging .....	10-3
Sample build.xml File .....	10-4
<b>Invoking the OC4J Ant Tasks .....</b>	<b>10-4</b>
Setting the Deployer URI .....	10-5
Invoking a Task on a Group of OC4J Instances Within a Cluster .....	10-5
Invoking a Task on a Specific OC4J Instance Within a Cluster.....	10-6
Invoking a Task on a Standalone OC4J Server .....	10-6
Deploying an EAR File .....	10-7
Deploying a Standalone WAR File.....	10-8
Deploying a Standalone RAR File .....	10-9
Redeploying an Archive.....	10-10
Binding All Web Modules to a Single Web Site .....	10-11
Binding a Specific Web Module to a Specific Web Site and Setting the Context URI .....	10-11
Undeploying an Archive.....	10-12
Updating Modified Classes Only in a Deployed EJB Module .....	10-12

Installing a Shared Library .....	10-13
Modifying an Existing Shared Library.....	10-14
Starting or Stopping an Application .....	10-16

## 11 Deploying with the admin\_client.jar Utility

<b>Overview of admin_client.jar Usage .....</b>	<b>11-1</b>
Understanding the admin_client.jar Syntax and URI Specification .....	11-2
Deploying to a Group of OC4J Instances Within a Cluster .....	11-2
Deploying to a Specific OC4J Instance.....	11-3
Deploying to a Standalone OC4J Server .....	11-3
Validating a URI.....	11-4
Printing Usage Text to the Console .....	11-4
Enabling Logging .....	11-4
<b>Deploying an Archive .....</b>	<b>11-5</b>
Deploying a J2EE Application (EAR).....	11-5
Deploying a Standalone Web Module (WAR).....	11-7
Deploying a Standalone Resource Adapter (RAR) .....	11-8
<b>Binding Web Modules to a Web Site Post-Deployment.....</b>	<b>11-9</b>
Binding All Web Modules to a Single Web Site .....	11-9
Binding a Specific Web Module to a Specific Web Site and Setting the Context Root .....	11-9
<b>Redeploying an Archive .....</b>	<b>11-10</b>
<b>Undeploying an Archive.....</b>	<b>11-11</b>
Undeploying an EAR or Standalone WAR .....	11-11
Undeploying a Standalone RAR.....	11-11
<b>Updating Modified Classes Only in a Deployed EJB Module .....</b>	<b>11-11</b>
<b>Creating and Managing Shared Libraries .....</b>	<b>11-12</b>
Installing a Shared Library .....	11-12
Modifying an Existing Shared Library.....	11-13
Viewing the Contents of a Shared Library .....	11-14
List All Shared Libraries.....	11-14
<b>Starting/Stopping/Restarting an Application.....</b>	<b>11-14</b>

## 12 Deploying to Standalone OC4J with admin.jar

Understanding the admin.jar Syntax .....	12-1
Deploying/Redeploying an Application .....	12-2
Undeploying an Application .....	12-4
Updating an EJB Module Within a Deployed Application .....	12-4
Deploying/Redeploying a Standalone Connector.....	12-5
Undeploying a Standalone Connector .....	12-6

## 13 Deploying Web Applications from Eclipse

Installing the Web Tools Platform .....	13-1
Deploying a Web Application with the Web Tools Platform .....	13-1
Connecting to OC4J from Eclipse .....	13-1
Building a Web Application .....	13-2
Deploying a Web Application.....	13-2

<b>14</b>	<b>Automatic Deployment in OC4J</b>	
	What Is Automatic Deployment in OC4J? .....	14-1
	When Should Automatic Deployment Be Used? .....	14-1
	Using an "Auto-Deployment" Directory .....	14-2
	Using the Check For Updates Feature .....	14-2
	Enabling/Disabling Check For Updates .....	14-3
	What Types of Files Are Automatically Redeployed? .....	14-3
	What Happens During Redeployment? .....	14-4
	Forcing a One-Time Redeployment Using admin.jar .....	14-5
<b>15</b>	<b>Troubleshooting Deployment Errors</b>	
	Interruptions During Application Deployment .....	15-1
	Exceptions During Application Deployment .....	15-1
	OC4J Out of Memory Errors .....	15-1
	Java Compiler Out of Memory Errors .....	15-2
	Stack Overflow Errors .....	15-2
	"Too many open files" Errors .....	15-2
<b>A</b>	<b>Third Party Licenses</b>	
	ANTLR .....	A-1
	The ANTLR License .....	A-1
	Apache .....	A-1
	The Apache Software License .....	A-2
	Apache SOAP .....	A-6
	Apache SOAP License .....	A-7
	DBI Module .....	A-10
	Perl Artistic License .....	A-10
	Preamble .....	A-10
	Definitions .....	A-10
	FastCGI .....	A-12
	FastCGI Developer's Kit License .....	A-12
	Module mod_fastcgi License .....	A-13
	Info-ZIP Unzip Package .....	A-13
	The Info-ZIP Unzip Package License .....	A-14
	JSR 110 .....	A-14
	Jaxen .....	A-14
	The Jaxen License .....	A-14
	JGroups .....	A-15
	The GNU License .....	A-15
	mod_mm and mod_ssl .....	A-22
	OpenSSL .....	A-23
	OpenSSL License .....	A-23
	Perl .....	A-25
	Perl Kit Readme .....	A-25
	mod_perl 1.29 License .....	A-26
	mod_perl 1.99_16 License .....	A-27

Perl Artistic License .....	A-30
Preamble.....	A-30
Definitions.....	A-30
<b>SAXPath</b> .....	A-32
The SAXPath License.....	A-32
<b>W3C DOM</b> .....	A-33
The W3C License.....	A-33

## Index



---

---

# Preface

This book covers various aspects of deploying J2EE-compliant applications and standalone modules into Oracle Containers for J2EE 10g Release 3 (10.1.3).

This preface contains the following sections:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

## Intended Audience

This document is intended for the following audiences:

- A "professional services" person engaged in deploying applications into OC4J
- A systems administrator responsible for configuring and administering an OC4J installation
- A developer or architect involved in creating or designing a J2EE application who wants to avoid design pitfalls that could cause deployment and scalability problems

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

- General Web technology
- The J2EE environment
- General system administration

## 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 Configuration and Administration Guide*  
This discusses how to configure and administer applications for OC4J, including use of the Oracle Enterprise Manager 10g Application Server Control Console, use of standards-compliant MBeans provided with OC4J, and, where appropriate, direct use of OC4J-specific XML configuration files.
- *Oracle Containers for J2EE Configuration and Administration Guide*  
This discusses how to configure and administer applications for OC4J, including use of the Oracle Enterprise Manager 10g Application Server Control Console, use of standards-compliant MBeans provided with OC4J, and, where appropriate, direct use of OC4J-specific XML configuration files.
- *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:

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



---

---

# Getting Started

This chapter provides an introduction to deploying J2EE-compliant applications and standalone modules into OC4J. It includes the following topics:

- [Understanding Deployment in OC4J](#)
- [Options for Deploying into OC4J](#)

## Understanding Deployment in OC4J

As a J2EE 1.4 -compliant container, OC4J provides a J2EE-compliant infrastructure for deploying, undeploying and redeploying J2EE-compliant applications and modules.

A number of deployment options are provided with OC4J. See "[Options for Deploying into OC4J](#)" on page 1-2 for an overview of these tools.

## Valid Components for Deployment

Components that can be deployed into an OC4J instance include:

- A Web application packaged as a Web Application Archive (WAR) file
- Standalone modules packaged as Java Archive files (JARs) containing Web Services, Enterprise JavaBeans (EJB JAR), application clients (CAR) or resource adapters (RAR)
- A complete J2EE application packaged as an Enterprise Archive (EAR) file, which may contain zero or more of the archives listed above

All J2EE-compliant archive files deployed into OC4J must be packaged in accordance with the guidelines specified in the J2EE 1.4 specification. This includes packaging the J2EE standard deployment descriptors required for each type of component, such as J2EE Application Descriptor (`application.xml`) for applications and the J2EE Web Descriptor (`web.xml`) for Web modules.

See the "Application Assembly and Deployment" chapter of the *Java 2 Platform Enterprise Edition Specification, version 1.4* for details.

## Support for the J2EE Application Deployment API (JSR-88) in OC4J

The OC4J deployment infrastructure implements the functionality outlined in 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.

Specifically, the JSR-88 compliant features in OC4J provide the ability to:

- Start an application immediately upon deployment, making it available to clients

- Stop an application, making it unavailable to clients
- Undeploy an application or module
- Redeploy an application or module, essentially updating the currently installed application with an updated version
- Create a *deployment plan* containing the aggregated OC4J-specific configuration data needed to deploy a component into OC4J. See [Chapter 8, "Working With Deployment Plans"](#), for details on the JSR-88 implementation in OC4J.

## “Hot Deployment” in OC4J

The term *hot deployment* refers to the process of deploying archive files - EARs, WARs, JARs, etc. - and their associated XML descriptor files on a production application server without shutting down or re-starting (bouncing) the server.

Hot deployment or redeployment of an application or standalone module into OC4J is generally supported as long as no changes are made during the deployment process to existing data source, JMS or RMI configuration files, and the structure of an Enterprise JavaBean replacing an existing EJB has not changed.

In addition, libraries at the container level cannot be deployed in this manner. If an application is dependent upon a newer library, OC4J must be restarted.

See "[Redeploying an Application](#)" on page 2-4 for details on redeploying applications to an OC4J instance.

## Options for Deploying into OC4J

You have a number of options for deploying applications into OC4J, including utilities packaged with OC4J.

- [Application Server Control Console User Interface](#)
- [OC4J Ant Tasks](#)
- [admin\\_client.jar Command Line Utility](#)
- [admin.jar Command Line Utility](#)
- [Oracle JDeveloper](#)

Most of these options allow you to deploy an application to a "group" of instances within an Oracle Application Server cluster. A *group* is defined as a loosely synchronized 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 an application can be deployed.

## Application Server Control Console User Interface

The *Oracle Enterprise Manager 10g Application Server Control Console* provides a Web-based user interface for completing deployment-related tasks, including:

- Deploying, undeploying and redeploying standalone modules (WARs, CARs, etc.) into the selected OC4J instance
- Creating and editing re-usable deployment plans
- Setting application-specific security and application clustering configurations
- Importing shared libraries for use by the application, as well as modifying the list of inherited libraries

- Deploying to a specific OC4J instance or to a group of instances within an Oracle Application Server cluster

Because it provides such robust functionality, Application Server Control Console should be your first choice for performing deployment-related tasks.

See [Chapter 9, "Deploying with Application Server Control Console"](#) for details.

## OC4J Ant Tasks

OC4J includes a set of Ant tasks for executing deployment tasks on an OPMN-managed OC4J instance, a standalone OC4J server or across all OC4J instances within a cluster. These tasks provide another option for scripting the deployment process.

Specifically, you can:

- Deploy, undeploy or redeploy an application (EAR), a standalone Web module (WAR) or a standalone resource adapter (RAR) to a specific OC4J instance or to a group of instances within a cluster
- Incrementally update a deployed EJB module with modified classes
- Stop or start a specific application, on a specific OC4J instance or cluster-wide

See [Chapter 10, "Deploying with the OC4J Ant Tasks"](#) for an overview of the deployment-specific Ant tasks and guidelines for integrating the tasks into your application build process.

## admin\_client.jar Command Line Utility

The `admin_client.jar` tool provided with OC4J can be used to execute deployment tasks to an OPMN-managed OC4J instance, to a standalone OC4J server or across all OC4J instances within a cluster. Specifically, you can:

- Deploy, undeploy or redeploy an application (EAR), a standalone Web module (WAR) or a standalone resource adapter (RAR) to a specific OC4J instance or to a group of instances within a cluster
- Incrementally update a deployed EJB module with modified classes
- Stop or start a specific application, on a specific OC4J instance or cluster-wide

See [Chapter 11, "Deploying with the admin\\_client.jar Utility"](#) for instructions on using this tool.

## admin.jar Command Line Utility

The `admin.jar` command-line utility provided with OC4J can be used to deploy applications to a standalone OC4J server only. It cannot be used to deploy applications to an OPMN-managed OC4J instance.

- Deploy, undeploy and redeploy a J2EE application packaged within an EAR file
- Deploy, undeploy and redeploy a standalone resource adapter packaged within an EAR file
- Update an EJB module within a deployed application

Note that deployment of a standalone Web module packaged in a WAR file is not supported.

See [Chapter 12, "Deploying to Standalone OC4J with admin.jar"](#) for instructions on deploying applications with this tool.

## Oracle JDeveloper

Oracle JDeveloper 10g is a J2EE integrated development environment with end-to-end support for developing, debugging, and deploying e-business applications and Web services.

JDeveloper allows you to build J2EE applications and Web services from scratch, or jump-start the process by beginning with a J2EE framework. Whichever approach you prefer, JDeveloper offers a full suite of productivity tools to support your work from start to finish.

JDeveloper provides the ability to deploy a J2EE application into an OC4J instance directly from within the project structure. It also allows a deployment plan to be created and optionally saved as an XML file.

See the online help provided with JDeveloper for instructions on deploying applications.



---

---

## Deploying A J2EE Application Into OC4J

This chapter describes the processes of deploying and undeploying a J2EE application packaged within an EAR file into an OC4J instance. It includes the following sections:

- [Overview of the Application Deployment Process](#)
- [What Happens During Deployment?](#)
- [Redeploying an Application](#)
- [Undeploying an Application](#)

### Overview of the Application Deployment Process

OC4J provides a streamlined, user-friendly deployment process. A J2EE-compliant EAR file can be deployed as-is, with no changes or repackaging required. You simply have to point to the location of the EAR file; OC4J will automatically deploy the modules that comprise the application, including EJB JAR and WAR files.

The following section outlines the general steps involved in deploying an EAR into OC4J. The following topics are covered:

- [Designating a Parent Application](#)
- [Binding the Web Application to a Web Site](#)
- [Creating or Applying a Deployment Plan](#)
- [Dynamic HTTP Server Mount Point Configuration in Oracle Application Server](#)

---

---

**Note:** Deploying an EAR from a read-only shared directory is not recommended, as errors may occur. Copy the EAR file to a local directory first, then deploy it.

---

---

### Designating a Parent Application

Every application deployed into OC4J must have a designated parent application. The default parent is the *global application* packaged with OC4J, which is named `default`.

Designating an application as a parent enables classes and services to be shared among the child applications. A child application sees the namespace of its parent application, and inherits the set of shared libraries imported by the parent. Configuration data is also imported from the parent, although it can be overridden at the child application level.

Once deployed, any method within the child application can invoke any method within the parent application. This is a means to enable methods in one JAR to see

EJBs that have been deployed in another JAR. This is useful for deploying all service EJBs in a single JAR file, where its users declare the service application as its parent.

## Binding the Web Application to a Web Site

A Web application deployed as part of a J2EE application must be bound to the Web site that will be used to access it. This binding is accomplished by specifying the *name* portion of the *name-web-site.xml* configuration file that defines the Web site to bind the Web application to.

In most cases, applications will be bound to the default Web site, which is defined by the *default-web-site.xml* configuration file and listens for requests on port 8888. All Web site configuration files, including *default-web-site.xml*, are stored in the `ORACLE_HOME/j2ee/home/config` directory.

The Web module context root, which will be appended to the URL used to access the application through a Web browser, is also set as part of this Web application enablement process. This value will typically be read from the *application.xml* deployment descriptor packaged with the application.

Note that if Oracle HTTP Server is used, the context root value will be used in the mount point definition used to route incoming Web requests to an appropriate OC4J instance. See ["Dynamic HTTP Server Mount Point Configuration in Oracle Application Server"](#) on page 2-2 for details.

## Creating or Applying a Deployment Plan

If deploying through the Oracle Enterprise Manager 10g Application Server Control Console, you will apply a *deployment plan*—a client-side aggregation of all the configuration data needed to deploy an archive into OC4J—as the final step of the deployment process. You can either create a new deployment plan for the application, or re-use an existing plan, which is especially useful during redeployment.

See [Chapter 8, "Working With Deployment Plans"](#) for more information on creating and using deployment plans.

## Dynamic HTTP Server Mount Point Configuration in Oracle Application Server

In a configuration in which Oracle HTTP Server (OHS) is used, Web requests are received via an OHS instance, which then routes the requests to an OC4J instance serving the requested application.

To route 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.

Prior to Oracle Application Server Release 3 (10.1.3), configuration of these application-specific mount points was completely manual. When a new application was deployed to an OC4J instance, a new mount point had to be manually added to `mod_oc4j.conf`, the configuration file for the `mod_oc4j` module within OHS which forwards requests to OC4J instances. The OHS instance then had to be restarted.

In Oracle Application Server 10.1.3, mount point configuration is completely automated, eliminating the need for manual file configuration or OHS restarts. Every OC4J instance within the cluster sends mount point data for each of its deployed applications to OHS, which adds this information to its internal routing table.

When a new application is deployed to an OC4J instance, its mount point information is transmitted to OHS, enabling OHS to dynamically "discover" the application. Mount point information includes:

- The OC4J host address
- 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 application name  
This value is defined in the `*-web-site.xml` configuration file for the Web site the application is bound to.
- The Web context(s) defined for the application  
This value is also set in the `*-web-site.xml` configuration file.

The sending and receiving of mount point notifications is managed by Oracle Notification Server (ONS), a component of Oracle Process Manager and Notification Server (OPMN) that is installed by default with every OC4J and OHS instance in an Oracle Application Server configuration

## What Happens During Deployment?

When you deploy an application packaged within an EAR, the following occurs:

1. If the application is being redeployed, the existing installation is first undeployed from OC4J.
2. OC4J copies the EAR file to the “master” deployment directory, which defaults to the `ORACLE_HOME/j2ee/home/applications/` directory.
3. OC4J opens and parses the `application.xml` file packaged within the EAR file. This file is a standard J2EE descriptor that lists all of the modules contained within the EAR file. OC4J notes these modules and initializes the EAR environment.
4. OC4J reads the module deployment descriptors for each module type - Web module (WAR), EJB module, connector module, or client module - into memory. The JAR and WAR file environments are also initialized.
5. OC4J reacts to the configuration details contained in both the J2EE deployment descriptors and any OC4J-specific deployment descriptors. OC4J notes any J2EE component configurations that require action by OC4J, such as wrapping EJBs with their interfaces.
6. OC4J writes out new OC4J-specific configuration files to the `ORACLE_HOME/j2ee/home/application-deployments/app_name` directory, according to the contents of the deployment plan. Note that if one or more OC4J-specific deployment descriptors were supplied, you may notice that OC4J added additional elements to the generated files.

Any generated classes, such as EJB interface wrapper classes, are compiled and put into new subdirectories of this directory. For example, EJB wrapper classes are generated within an archive named `deployment-cache.jar` within the `ORACLE_HOME/j2ee/home/application-deployments/app_name/jar_name.jar/` directory, where `jar_name.jar` corresponds to the name of a deployed EJB JAR.

7. Finally, OC4J updates the OC4J `server.xml` configuration file with the notation that this application has been deployed.

## Example of a Deployed Application Directory Structure

The following example shows the key areas of the exploded directory structure created when an archive named `utility.ear` is deployed, assuming default settings for the target directories. The EAR includes a Web module (`utility_web.war`) and an EJB JAR (`utility_ejb.jar`) containing a single stateful session EJB.

Note how OC4J cleanly separates the standard J2EE content and the OC4J-specific files within the exploded directory structure. The original archives and the standard J2EE descriptors are copied to the `oc4j/j2ee/home/applications/` directory, enabling these files to be used in a redeployment of the application. The OC4J-specific descriptors generated during deployment are written to the `oc4j/j2ee/home/application-deployments/` directory.

Also note the EJB wrapper class, `UtilityManager_StatefulSessionBeanWrapper.class`, generated within the `deployment-cache.jar` archive. During deployment, OC4J generates a wrapper class for each EJB packaged within an EJB JAR. The wrapper classes generated for the EJBs within an EJB JAR are contained within an archive named `deployment-cache.jar`, which is in turn contained within a generated JAR file with the same name as the deployed EJB JAR.

```
j2ee/home/
  application-deployments/
    utility/
      orion-application.xml
      utility_web/
        orion-web.xml
      utility_ejb.jar/
        orion-ejb-jar.xml
        deployment-cache.jar/
          UtilityManager_StatefulSessionBeanWrapper.class
  applications/
    utility.ear
    utility/
      utility_web.war
      utility_ejb.jar
      META-INF/
        application.xml
      utility_web/
        index.html
      META-INF/
      WEB-INF/
        web.xml
        classes/
          Example.class
```

## Redeploying an Application

Redeploying a J2EE application packaged within an EAR file prompts OC4J to undeploy the previous instance of the J2EE application, including any embedded resource adapters packaged with the application. Therefore, there is no need to undeploy the application before redeploying.

If you saved the deployment plan from the previous deployment, you can re-use it during the redeployment. By default, the deployment plan will be initialized using the existing application configuration and applied to the redeployment. The previously generated OC4J descriptors will be overwritten based on the content of the deployment plan.

If changes are made to any of the following configuration files as part of redeployment, OC4J must be restarted for the changes to take effect. Otherwise, a restart of OC4J is not required after redeploying an application. Note that this requirement applies only to configuration files that are already installed on the server.

- `data-sources.xml`
- `jms.xml`
- `rmi.xml`

The application is completely inaccessible during redeployment, and incoming requests will not be processed until the updated application is restarted by OC4J when deployment is complete.

## Impact of Redeploying a Parent Application

After redeploying an application that is the parent of one or more child applications, you should ideally restart each child application. Restarting will ensure that the child applications are able to access any inherited classes or shared libraries provided through the parent.

## Undeploying an Application

An application can be removed from an OC4J instance using the following:

- The **Applications** section of the Application Server Control Console user interface.
- The `-undeploy` option provided through the `admin_client.jar` command line utility. See "[Undeploying an Archive](#)" on page 11-11 for instructions on using this option.

Removing a J2EE application from an OC4J instance results in the following:

- The application is removed from the OC4J runtime.
- All bindings for the Web applications are removed from all the Web sites to which the Web modules were bound.
- All application files are removed from both the `applications/` and `application-deployments/` directories.

## Impact of Undeploying a Parent Application

When an application that is the parent of one or more child applications is undeployed, the child applications are also undeployed.

This means that all of the related applications - the parent as well as its dependent applications - must be redeployed. Bear this in mind before undeploying a parent application.



---

---

## Deploying Large Applications

This chapter provides guidelines for configuring OC4J and the Java compiler for deployment of large J2EE applications containing large numbers (~100) of Enterprise JavaBeans.

This chapter includes the following sections:

- [Specifying the Compilation Mode to Use](#)
- [Configuring the Java Compiler](#)
- [Tuning the OC4J JVM for Large Deployments](#)

### Specifying the Compilation Mode to Use

When an application containing EJB JAR files is deployed, OC4J automatically generates a wrapper class for each EJB that implements the various component interfaces packaged with the EJB application. OC4J then invokes the Java compiler to compile these generated EJB wrapper classes.

OC4J supports two modes for compiling EJB wrapper classes: Batch mode and non-batch mode. Each is discussed below.

#### Batch Mode

This is the default compilation mode used by OC4J. In general, batch mode provides faster time to deployment when deploying large EJB-heavy applications. However, it also requires a greater heap memory allocation than non-batch mode deployment.

In this mode, OC4J makes a single call to the Java compiler to compile all of the Java wrapper code for all of the EJBs within the EAR being deployed.

If the compiler is configured to run in out-of-process mode - which it is by default in OC4J - then OC4J will create a single JVM process to execute compilation of the wrapper code. See "[Configuring Out-of-process Versus In-process Compiler Execution](#)" on page 3-2 for instructions on configuring the compiler to run out-of-process or in the same JVM process as OC4J.

#### Non-Batch Mode

If you find that OC4J throws `java.lang.OutOfMemory` exceptions while compiling in batch mode, you may want to compile in non-batch mode instead, as it requires less memory allocation. However, using this mode will result in a longer time to deployment.

In this mode, OC4J makes multiple calls to the compiler—one for each EJB JAR file within the EAR being deployed.

If the Java compiler is configured to run in out-of-process mode, OC4J will create a JVM process for each EJB JAR file included in the EAR file being deployed.

To deploy in non-batch mode, set the `batch-compile` attribute of the `<orion-application>` element in `application.xml` or `orion-application.xml` to `false`. For example:

```
<orion-application ... batch-compile="false" .../>
```

---

---

**Note:** The `ejbdeploy.batch` system property, which had previously been used to specify batch versus non-batch compilation mode, is de-supported in this release of OC4J.

---

---

## Configuring the Java Compiler

This section provides guidelines on configuring the Java compiler that will compile the EJB wrapper classes, JSPs during deployment. It includes the following topics:

- [Specifying an Alternative Java Compiler](#)
- [Configuring Out-of-process Versus In-process Compiler Execution](#)
- [Summary of Java Compiler Configuration Parameters](#)

Note that all compiler configuration parameters are specified as attributes of the `<java-compiler>` element in `ORACLE_HOME/j2ee/instance/config/server.xml`, the OC4J server configuration file.

### Specifying an Alternative Java Compiler

By default, OC4J uses the `javac` compiler packaged with the Sun Microsystems JDK to compile generated EJB wrapper classes, JavaServer Pages, Web Services classes and any `.java` files packaged with an application. However, you can configure OC4J to use a different Java compiler by modifying the `<java-compiler>` element in `server.xml` with the alternative compiler configuration.

For example, the following notation will cause OC4J to use the Jikes compiler from IBM:

```
<java-compiler name="jikes" in-process="false"/>
```

### Configuring Out-of-process Versus In-process Compiler Execution

The Java compiler can be configured within OC4J to execute in one of two modes: out-of-process or in-process.

#### Out-of-process

In this mode, 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. Once compilation of the EJB wrapper classes is complete, the memory allocated to the associated JVM will be released and made available to other processes.

Set the `in-process` attribute of the `<java-compiler>` element to `false` to execute the compiler in this mode.



**In-process**

In this mode, the compiler executes within the same JVM process as OC4J. Set the `in-process` attribute of the `<java-compiler>` element to `true` to execute in this mode.

**Summary of Java Compiler Configuration Parameters**

Table 3–1 summarizes the attributes of the `<java-compiler>` element, which defines the Java compiler configuration in `server.xml`, the OC4J configuration file.

**Table 3–1 Attributes of the `<java-compiler>` Element**

Attribute	Description
<code>name</code>	Value: string Specifies the compiler name. Set to <code>javac</code> to use the Sun <code>javac</code> compiler.
<code>bindir</code>	Value: string Specifies the absolute path to the compiler directory. This attribute does not need to be specified to use the default <code>javac</code> compiler.
<code>in-process</code>	Value: Boolean Default: <code>false</code> Indicates whether to execute the Java compiler in-process ( <code>true</code> ) or out-of-process ( <code>false</code> ).
<code>options</code>	Used to pass command line options to the Java compiler. Separate multiple options with a space. For example: <code>&lt;java-compiler name="javac" options="-J-Xmx2048m -J-Xss=8m"/&gt;</code>
<code>encoding</code>	Value: string Default: <code>ISO-8859-1</code> Specifies the source file encoding to use.
<code>extdirs</code>	Indicates the compiler extension library location.
<code>debug</code>	Value: Boolean Default: <code>false</code> Set to <code>true</code> to generate compilation-time debugging output.

**Tuning the OC4J JVM for Large Deployments**

In addition to setting the OC4J JVM heap size (young and old generations), you should also pay attention to the permanent generation size when deploying large applications.

The JVM permanent generation plays an important role when you are deploying large applications. Since a large application may contain hundreds or even thousands of EJBs, the OC4J JVM needs to load a large number of classes, which requires a higher value for permanent generation size.

If the permanent generation size is set too low, you may see `java.lang.OutOfMemoryError` errors from the JDK even if you have plenty of free memory in the heap. If this occurs, you can increase the permanent generation (Perm) size by setting the `-XX:MaxPermSize` property.

For example, to set the total heap size to 768 MB, set the following at OC4J startup:

```
java -Xms512m -Xmx512m -XX:MaxPermSize=256m -jar oc4j.jar
```

To determine the value for the `-XX:MaxPermSize` property - including young and old generation sizes - you can use `visualgc` to monitor the OC4J JVM during application deployment. The `visualgc` garbage collection monitoring tool is included with the free `jvmstat 3.0` distribution available from Sun Microsystems. See the following link for more information and to download the tool:

<http://java.sun.com/performance/jvmstat/>

---

---

## Deploying Enterprise JavaBeans

The following topics discuss deployment/redeployment of Enterprise JavaBean modules into an application running in an OC4J instance.

- [Overview of EJB Deployment](#)
- [Generating Client-side IIOP Stubs](#)
- [Incremental Redeployment of Updated EJB Modules](#)
- [Impact of EJB Redeployment on Application Clients](#)

### Overview of EJB Deployment

The EJB deployment process is highly automated. When an application containing one or more EJB JAR files is deployed, OC4J executes as follows:

1. OC4J generates a wrapper class for each of the home interfaces (EJBHome and EJBLocalHome implementations) and component interfaces (EJBObject and EJBLocalObject implementations) packaged within each EJB JAR file.
2. OC4J invokes the Java compiler that it is configured to use to compile the generated EJB wrapper classes. The compiled classes are output to an archive named `deployment-cache.jar` in a new subdirectory with the same name as the deployed EJB JAR in `ORACLE_HOME/j2ee/instance/app_name/application-deployments/`.

For example, suppose you deployed `mystore.ear`, which contains `inventory-ejb.jar`. The compiled wrapper classes will be generated in `ORACLE_HOME/j2ee/instance/mystore/application-deployments/inventory-ejb/deployment-cache.jar`.

3. OC4J optionally generates client-side IIOP stubs for each home and component interface if configured to do so.

See "[Example of a Deployed Application Directory Structure](#)" on page 2-4 for an example of the directory structure created for an application by OC4J.

Note that because of the amount of processing performed by OC4J and the Java compiler, deploying an EJB JAR file containing a large number (~100) of EJBs may significantly increase the amount of time required to deploy an application. See [Chapter 3, "Deploying Large Applications"](#) for guidelines on tuning OC4J and the Java compiler for large application deployment.

## Generating Client-side IIOP Stubs

OC4J optionally generates client-side IIOP stubs for each home and component interface if configured to do so when deploying with Application Server Control Console, `admin_client.jar` or the OC4J Ant tasks.

- [Generating Stubs with Application Server Control](#)
- [Generating Stubs with `admin\_client.jar`](#)
- [Generating Stubs with the OC4J Ant Tasks](#)

### Generating Stubs with Application Server Control

See "[Deploying/Redeploying an Application](#)" on page 9-3 for instructions on deploying applications with Application Server Control Console.

1. Before deploying the EJBs, configure OC4J to generate client-side IIOP stubs in Application Server Control Console.
  - Click the **Administration** tab.
  - Check the **Generate IIOP Client Stubs when Compiling EJBs** box under Compile Time Parameters.
2. Next, enable stub generation at deployment time:
  - In the third panel (Deployment Settings) of the deployment wizard, click **Edit Deployment Plan**.
  - Set `enableIIOP` to `true`.

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/app_name` 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/app_name/ejbModuleName/` directory.

### Generating Stubs with `admin_client.jar`

The `-deploy` command on `admin_client.jar` provides two options for generating IIOP stubs: One for generating stubs on the server, the other for generating stubs on the server and copying the new stubs to another location.

1. Before deploying the EJBs, set the `-DGenerateIIOP` system property, which configures OC4J to generate client-side IIOP stubs at startup.
  - In standalone OC4J, specify this system property on the OC4J command line:

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

- In an OPMN-managed OC4J instance, set the property in `opmn.xml`:

```
<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>
```

2. Next, deploy the application with the `admin_client.jar -deploy` command. See ["Deploying an Archive"](#) on page 11-5 for details on using this command.
  - Include `-enableIIOP` 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.
  - Include `-iiopClientJar path` to generate stubs in the same locations specified above, as well as to the path specified with the switch.

## Generating Stubs with the OC4J Ant Tasks

The `deploy` Ant task also provides two options for generating IIOP stubs: One for generating stubs on the server, the other for generating stubs on the server and copying the new stubs to another location.

1. Before deploying the EJBs, set the `-DGenerateIIOP` system property, which configures OC4J to generate client-side IIOP stubs at startup. See ["Generating Stubs with admin\\_client.jar"](#) above for details.
2. Next, deploy the application with the `deploy` task. See ["Deploying an EAR File"](#) on page 10-7 for details on using this task.
  - Include `enableIIOP` 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.
  - Include `iiopClientJarPath` to generate stubs in the same locations specified above, as well as to the path specified with the switch.

## Incremental Redeployment of Updated EJB Modules

OC4J supports incremental or partial redeployment of EJB modules that are part of a deployed application. This feature makes it possible to redeploy only those beans within an EJB JAR that have changed to be deployed, without requiring the entire module to be redeployed. Previously deployed beans that have not been changed will continue to be used.

This functionality represents a significant enhancement over previous releases of OC4J, which treated an EJB module as a single unit, requiring that the module first be undeployed, then redeployed with any updates.

The `-updateEJBModule` command in both the `admin.jar` and `admin_client.jar` utilities can be used to incrementally update a deployed application with one or more EJBs contained within an EJB JAR file.

A restart of OC4J is required only if changes are made to the EJB configuration data during the redeployment process. If no changes are made, a “hot deployment” can be performed without re-starting OC4J.

The incremental redeployment operation will automatically stop the application containing the EJB(s) to be updated, then automatically restart the application when finished.

---

---

**Note:** During redeployment, any client connections to the EJB being updated will be lost. It is strongly recommended that the application be stopped before redeploying the EJB. All existing requests will be allowed to complete, but no new requests will be allowed until the application is restarted.

---

---

## Impact of EJB Redeployment on Application Clients

This section discusses the impact of EJB redeployment on existing clients, which differs depending on the type of EJBs used in the application. It includes the following topics:

- [Redeploying Session Beans](#)
- [Redeploying Entity Beans](#)

### Redeploying Session Beans

The following describe issues related to redeploying session beans.

#### Stateless Session Beans

For applications that include stateless session EJBs, the redeployment appears seamless to users, with no interruption in service. Existing requests will be served by current bean instances, while new requests will be served with new instances.

#### Stateful Session Beans

For an application or Web service that utilizes active stateful session EJB instances, you must explicitly specify a “persistence directory” where client state data will be persisted through serialization during the undeployment and redeployment processes. If this directory is not specified, all serialized state data will be lost during undeployment.

The persistence directory is defined in the <persistence> element within the `orion-application.xml` configuration file. This directory can also be set as the value of the `persistencePath` property through the deployment plan editor at the time the EJB archive is deployed. See [“Setting Web Module Configuration Properties”](#) on page 8-11 for information.

Also note that for existing clients, a serialization-related exception will occur if there are any changes in the structure of the session beans deployed with the module.

### Redeploying Entity Beans

The following describe issues related to redeploying entity beans.

### **Bean-Managed Persistence Beans**

The application developer is responsible for handling any exceptions that may occur as a result of "hot deployment" of EJBs.

### **Container-Managed Persistence Beans**

An unknown number of side effects may likely occur if the structure, types and relationships of the container managed fields within an EJB are changed. For this reason, OC4J should always be re-started after you make any changes.





---

---

## Deploying Web Modules

This chapter discusses deploying and redeploying Web modules into an OC4J instance. It includes the following topics:

- [Deploying a Standalone Web Module](#)
- [Redeploying a Standalone Web Application](#)
- [Redeploying an Updated Web Module into an Existing Application](#)
- [Adding or Modifying JavaServer Pages in an Active Web Module](#)

### Deploying a Standalone Web Module

A standalone Web module packaged as a WAR file can be deployed into an OC4J instance. However, the Web module must be designated a child of either the `default` application (if a standalone Web application) or another deployed application that does not already contain a Web module component.

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. Repackage the WAR in the application's EAR file and redeploy the application instead.

OC4J wraps the standalone WAR file within a generated EAR file, then deploys the EAR to the default deployment directory, `ORACLE_HOME/j2ee/instance/applications`. The EAR includes a generated `application.xml` deployment descriptor, which includes the context root appended to the URL used to access the Web module.

A restart of OC4J is not required after deployment of a standalone Web module.

### Redeploying a Standalone Web Application

In this context, a standalone Web application is a WAR file that has the `default` application specified as its parent. When a standalone Web application is redeployed, OC4J performs the following:

- Removes the Web application from its execution space;
- Removes the classloader that was associated with execution of the Web application;
- Re-parses the application-specific `web.xml` and `orion-web.xml` descriptors to pick up any changes; and
- Re-initializes servlet listeners, filters and mappings.

Existing sessions are either purged or serialized out to a file in the persistence directory specified for the Web application, which is defined in the `persistence-path` attribute of the `<orion-web-app>` element. When the newly deployed application is started, it looks in the persistence directory for the file containing the serialized sessions.

---

---

**Notes:**

- If you update a servlet `.class` file under `/WEB-INF/classes`, then upon the next request, the servlet and its dependency classes are reloaded and the Web application is automatically redeployed if OC4J polling is enabled.
  - Redeployment does not significantly affect OC4J-specific descriptors such as `orion-application.xml` and `orion-web.xml` in the server deployment directory. After you trigger reloading, the previously copied or generated files will keep any previously specified non-default settings.
- 
- 

## Redeploying an Updated Web Module into an Existing Application

An updated Web module packaged in a WAR cannot be redeployed into a J2EE application running on OC4J. Instead, the WAR must be re-packaged within the application's EAR file, and the entire EAR must then be redeployed.

## Adding or Modifying JavaServer Pages in an Active Web Module

OC4J allows new JavaServer Pages (JSPs) to be added to an actively running Web module - as well as existing JSPs to be modified - without requiring an application redeployment or restart.

To use this feature, simply drop a new or updated JSP into the appropriate directory within the exploded WAR file structure in the OC4J instance, which is `ORACLE_HOME/j2ee/instance/applications/appName/webModuleName/`. OC4J will translate the page and load (or reload) it into the runtime.

This feature is enabled by default, and is managed through the **When a JSP Changes** JSP configuration parameter. This parameter can be set through the Application Server Control Console user interface as follows:

1. Click the **Administration** link -> **JSP Properties** icon.
2. Set the **When a JSP Changes** parameter to one of the following:
  - **Recompile JSP**  
This is the default setting. OC4J will check the timestamp of the JSP page, retranslate it and reload if it has been modified since it was last loaded. The functionality described below for **Reload Classes** will also be executed.
  - **Reload Classes**  
OC4J will check the timestamp of classes generated by the JSP translator, such as page implementation classes, and reload any that have changed or been redeployed since they were last loaded.

This might be useful, for example, when you deploy or redeploy compiled classes, but not JSP pages, from a development environment to a production environment.

- **Do Nothing**

Set this parameter to disable the auto-reload feature. New or updated JSPs will not be automatically loaded into the OC4J runtime, and the container will not perform any timestamp, any retranslating, of JSP pages or reloading of generated Java classes. This is the most efficient mode for a production environment, where JSP pages are not expected to change frequently.

3. Restart the OC4J instance for your changes to take effect.

This scenario assumes that any dependent classes already exist within the deployed Web module, and that the JSP updates do not require any changes to the `web.xml` or `orion-web.xml` configuration files. If either of these conditions is false, the Web module must be re-packaged as a WAR file and redeployed with the full application within an EAR file.

## Optionally Setting JSP Configuration Parameters in the XML Configuration File

In an OC4J standalone environment, you can allow new JSPs to be added to a running Web module by setting the `main_mode` `<init-param>` element directly in the `global-web-application.xml` file, which is the configuration file for the OC4J servlet and JSP containers. The values set through Application Server Control Console are persisted to this file.

Valid settings for `main_mode` are:

- `reload`: Maps to the **Reload Classes** parameter set through the Application Server Control Console user interface.
- `recompile` (default): Maps to the **Recompile JSP** parameter.
- `justrun`: Maps to the **Do Nothing** parameter.

The following example illustrates how to set the `main_mode` `<init-param>` element to `recompile`, which forces OC4J to check the timestamp of the JSP page, retranslate it, and reload it if it has been modified since it was last loaded.

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>oracle.jsp.runtimev2.JspServlet</servlet-class>
  <init-param>
    <param-name>precompile_check</param-name>
    <param-value>>true</param-value>
  </init-param>
  <init-param>
    <param-name>main_mode</param-name>
    <param-value>recompile</param-value>
  </init-param>
  <init-param>
    <param-name>javacmd</param-name>
    <param-value>javac -verbose</param-value>
  </init-param>
</servlet>
```



---

---

## Deploying Resource Adapters

A resource adapter can be packaged and deployed as a standalone RAR that is available as a shared resource to all applications within the OC4J instance.

This chapter includes the following topics:

- [Deploying a Standalone RAR](#)
- [Redeploying/Undeploying a Standalone RAR](#)

### Deploying a Standalone RAR

A resource adapter deployed as a standalone RAR is deployed as a child of the `default` application, making the connector available to all other applications deployed to the OC4J instance.

The resource adapter is deployed to the `ORACLE_HOME/j2ee/home/connectors` directory by default. This directory is specified in `ORACLE_HOME/j2ee/home/config/server.xml`, the OC4J server configuration file.

The following updates are also made to the OC4J instance:

- A new `<connector>` element defining the resource adapter is added to `ORACLE_HOME/j2ee/home/config/oc4j-connectors.xml`, which provides an enumeration of the standalone resource adapters deployed to the OC4J instance.
- An `oc4j-ra.xml` descriptor is generated in a new directory with the same name as the connector in the `ORACLE_HOME/j2ee/home/application-deployments` directory.

### Redeploying/Undeploying a Standalone RAR

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`.

Because of this requirement, redeploying or undeploying a standalone resource adapter is a task that will likely be rarely performed in a production environment.



---

---

## Deploying Web Services

This chapter discusses redeployment of Web services. It includes the following topics:

- [Deploying a Web Service](#)
- [When Should You Redeploy a Web Service?](#)

### Deploying a Web Service

A Web service can be packaged as a WAR or as an EJB JAR containing stateless session beans for deployment into OC4J.

Note that if an archive containing a Web service does not include a Web Services Description Language (WSDL) document, OC4J will generate a WSDL document at deployment time.

See [Chapter 5, "Deploying Web Modules"](#) for a discussion on deploying and redeploying WAR files into OC4J.

See [Chapter 4, "Deploying Enterprise JavaBeans"](#) for guidelines on deploying and redeploying EJB archives.

The deployment plan editor provided with Oracle Enterprise Manager 10g Application Server Control Console includes the ability to set values in the OC4J-specific Web services deployment descriptor (`oracle-webservices.xml`) at deployment time. See ["Setting Web Services Configuration Properties"](#) on page 8-25 for details.

### When Should You Redeploy a Web Service?

In general, the guidelines for updating a Web service in OC4J are the same as those for any WAR or EJB JAR containing stateless session beans.

However, redeployment of a Web service is essentially required when:

- Changes are made to the existing reliability configuration
- Changes are made to existing security policies or data set in XML configuration files

Redeployment is required so that an updated WSDL document containing the updated reliability and/or security notations can be supplied. Without the updated WSDL document, clients calling the Web service will not work correctly.

Ideally, a new WSDL document containing the updated reliability and/or security notations should be supplied with the WAR or EJB JAR during redeployment. If no

WSDL document is supplied, OC4J will generate a new document with the correct notations. However, data set in the previously deployed WSDL document will be lost.



---

---

## Working With Deployment Plans

This chapter provides instructions on creating and using deployment plans, which facilitate the process of editing and re-using configuration data when deploying archives into OC4J. It includes the following sections:

- [What Is a Deployment Plan?](#)
- [Creating or Editing a Deployment Plan](#)
- [Setting Properties in a Deployment Plan](#)

### What Is a Deployment Plan?

Like all J2EE containers, OC4J utilizes a number of vendor-specific deployment descriptor files that extend the standard J2EE deployment descriptors. For example, the OC4J-specific `orion-application.xml` descriptor extends the J2EE standard `application.xml` descriptor with configuration data specific to OC4J. See ["Overview of J2EE and OC4J Deployment Descriptors"](#) on page 8-2 for an overview of the relationship between J2EE and OC4J deployment descriptors.

A key feature of JSR-88 is the ability to create a *deployment plan* - a client-side aggregation of all the configuration data needed to deploy an archive into OC4J. This deployment plan can be edited at the time of deployment using the *deployment plan editor* functionality provided through the Application Server Control Console, providing a straightforward way to modify configuration data for a particular installation. (See ["Setting Properties in a Deployment Plan"](#) on page 8-4 for details.)

When the archive is deployed, both the archive and the deployment plan are sent to the OC4J server. OC4J uses the contents of the deployment plan to generate the various OC4J-specific descriptors within the `ORACLE_HOME/j2ee/instance/application-deployments` directory.

For example, if an EAR containing a WAR and an EJB JAR is deployed, the deployment plan will contain the aggregated configuration data for each of these archives. Upon deployment, this data would be written to the `orion-application.xml`, `orion-web.xml` and `orion-ejb-jar.xml` descriptors, respectively, generated by OC4J.

Once created, a deployment plan can optionally be saved as a file. It can then be re-used when redeploying the component or when deploying other components. If an existing deployment plan is not applied to a component at the time of deployment, a new plan is created by default.

**Note:** Deployment plans include additional properties that do not appear in a deployment descriptor, but are required by the deployment process. For example, a property specifying whether the EAR should be deleted after deployment has completed.

## How Do Deployment Plans Interact with Packaged Deployment Descriptors?

If one or more OC4J-specific descriptors - such as `orion-application.xml` or `orion-ejb-jar.xml` - are packaged within an archive being deployed, the deployment plan is initialized with the data within these files. The configuration data can then be edited through the deployment plan editor provided through the Application Server Control Console before deployment. (See "[Setting Properties in a Deployment Plan](#)" on page 8-4 for details.)

Note that changes made through the deployment plan editor are not written back to the archive. For example, suppose that the deployment plan editor is used to define a `UserManager` class to use at the application level. When the application is deployed, the `orion-application.xml` file generated within OC4J will contain the added `<user-manager>` element. The `orion-application.xml` file packaged within the archive will not.

## Overview of J2EE and OC4J Deployment Descriptors

Deployment descriptors are configuration files that are deployed with J2EE applications and modules. Each J2EE standard deployment descriptor is extended by a corresponding OC4J-specific descriptor. The following table provides a description of these files and illustrates how they relate to one another.

The XML Schema Definition (XSD) file that describes each OC4J-specific descriptor is also noted. You can view the current Oracle XSDs at the following link:

<http://www.oracle.com/technology/oracleas/schema/index.html>

**Table 8-1 J2EE and OC4J Deployment Descriptors**

J2EE Standard Descriptors	OC4J Proprietary Descriptors
<p><code>application.xml</code></p> <p>Specifies the components of a J2EE application, such as EJB and Web modules, and can specify additional configuration for the application as well. This descriptor must be included in the <code>/META-INF</code> directory of the application's EAR file.</p>	<p><code>orion-application.xml</code></p> <p>Generally defines OC4J-specific configurations such as security role mappings, data source definitions, JNDI namespace access and shared library replacements. Can also be used to specify additional modules, beyond those specified in the J2EE <code>application.xml</code> descriptor.</p> <p>The format of this file is defined by <code>orion-application-10_0.xsd</code>.</p>
<p><code>web.xml</code></p> <p>Specifies and configures a set of J2EE Web components, including static pages, servlets and JSP pages. It also specifies and configures other components, such as EJBs, that the Web components might call. The Web components might together form an independent Web application and be deployed in a standalone WAR file.</p>	<p><code>orion-web.xml</code></p> <p>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.</p> <p>The format of this file is defined by <code>orion-web-10_0.xsd</code>.</p>

**Table 8–1 (Cont.) J2EE and OC4J Deployment Descriptors**

J2EE Standard Descriptors	OC4J Proprietary Descriptors
<p><code>ejb-jar.xml</code></p> <p>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.</p>	<p><code>orion-<i>ejb-jar.xml</i></code></p> <p>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.</p> <p>The format of this file is defined in <code>orion-<i>ejb-jar-10_0.xsd</i></code>.</p>
<p><code>application-client.xml</code></p> <p>Describes the EJB modules and other resources used by a J2EE application client packaged in an archive.</p>	<p><code>orion-<i>application-client.xml</i></code></p> <p>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.</p> <p>The file format is defined in <code>orion-<i>application-client-10_0.xsd</i></code>.</p>
<p><code>ra.xml</code></p> <p>Contains information on implementation code, configuration properties and security settings for a resource adapter packaged within a RAR file.</p>	<p><code>oc4j-<i>ra.xml</i></code></p> <p>Contains deployment configuration data for a single resource adapter. This data includes such information as JNDI name to be used, EIS connection information, connection pooling parameters, and resource principal mappings.</p> <p>The file format is defined by <code>oc4j-<i>connector-factories-10_0.xsd</i></code>.</p> <p><code>oc4j-<i>connectors.xml</i></code></p> <p>In an OC4J instance with standalone resource adapters deployed, contains an enumeration of those standalone resource adapters. In a J2EE application with embedded resource adapters deployed, contains a list of embedded resource adapters that have been bundled with the application.</p> <p>This file is formatted according to <code>oc4j-<i>connectors-10_0.xsd</i></code>.</p>
<p><code>webservices.xml</code></p> <p>Describes a Web Service, including WSDL information and JAX-RPC mapping data, for a Web Service application packaged within a WAR file.</p>	<p><code>oracle-<i>webservices.xml</i></code></p> <p>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.</p> <p>This file is formatted according to <code>oracle-<i>webservices-10_0.xsd</i></code>.</p>

## Creating or Editing a Deployment Plan

Deployment plans can be created or edited through the deployment plan editor functionality available through the Web-based Oracle Enterprise Manager 10g Application Server Control Console interface and the J2EE and Studio Editions of the Oracle JDeveloper 10g integrated development environment.

- [Creating or Editing Deployment Plans with Application Server Control Console](#)
- [Creating or Editing Deployment Plans with Oracle JDeveloper](#)

## Creating or Editing Deployment Plans with Application Server Control Console

The deployment plan editor screens provided through the Oracle Enterprise Manager 10g Application Server Control Console are accessed through the third and final screen of the "deployment wizard". Simply click the **Edit Deployment Plan** button on this screen to view the editor.

Each archive being deployed is displayed as an XPath node in the left-hand navigation pane of the editor. Select an archive node to access the deployment plan properties for that archive type. You will be able to view the following:

- The J2EE standard descriptor packaged with the archive
- The current values for XML elements and attributes that will be set in the OC4J-specific descriptor generated at deployment time
- A set of properties that can be set in the deployment plan, each corresponding to an element or attribute in the OC4J-specific descriptor that will be generated at deployment time.

See "[Setting Properties in a Deployment Plan](#)" on page 8-4 for descriptions of each property that can be set. Note that not all parameters that can be set in an XML descriptor file can be set through the deployment editor; parameters set post-deployment are not exposed.

After creating a deployment plan, you can optionally save it as an XML file, enabling it to be applied to other application deployments.

You can also select an existing deployment plan to apply during a deployment in the first panel of the "deployment wizard". Once retrieved, a deployment plan can be edited or used as-is.

## Creating or Editing Deployment Plans with Oracle JDeveloper

The deployment plan editor functionality provided with Oracle JDeveloper 10g is presented as the **Configure Application** panel.

To use the plan editor, you must first create a connection of type **J2EE 1.4 Server** to the target OC4J instance. This is done through the wizard accessible via the Connection Navigator panel.

Once you are connected, the Configure Application panel is displayed after you select an archive and choose **Deploy to -> connection\_name**.

After creating a deployment plan, you can optionally save it as an XML file, enabling it to be applied to other application deployments.

See the "Deploying Applications" topic in the Oracle JDeveloper online help for instructions on deploying archives into OC4J.

## Setting Properties in a Deployment Plan

This section provides an overview of the properties that can be set in a deployment plan. It is organized based on the type of archive being deployed:

- [Setting J2EE Application Configuration Properties](#)
- [Setting Web Module Configuration Properties](#)
- [Setting Enterprise JavaBeans Module Configuration Properties](#)
- [Setting Web Services Configuration Properties](#)

- [Setting Application Client Configuration Properties](#)
- [Setting Resource Adapter Properties](#)

## Setting J2EE Application Configuration Properties

The following details the OC4J-specific properties that can be set when you are deploying a J2EE application packaged in an EAR file. Each property maps to an element attribute in the `orion-application.xml` descriptor.

### **applicationId**

Contains a string containing the application's unique identifier.

### **autoCreateTables**

Set to `true` to automatically create database tables for CMP Enterprise JavaBeans in this application. The default is `true`.

### **autoDeleteTables**

Specifies whether old database tables for CMP beans should automatically be deleted when redeploying this application. The default is `false`.

### **cluster**

Configures OC4J clustering at the application level. Clustering is typically configured at the global level; however, application-level settings will override the global configuration. See *Oracle Containers for J2EE Configuration and Administration Guide* for a detailed overview of the OC4J clustering framework.

- `enabled`: Specifies 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, meaning that clustering can be disabled for a specific application.
- `groupName`: 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.

- `allowColocation`: Specifies whether to allow application state to be replicated to a group member residing on the same host machine. The default is `true`.  
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.

- `writeQuota`: 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 member.
- `cacheMissDelay`: 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.

- `replicationPolicy`: Specifies the replication policy to apply, which defines when replication of data occurs.

The ideal values to set differ for Web modules and EJB components. See the *Oracle Containers for J2EE Configuration and Administration Guide* for valid values.

- `propertyConfig`: Contains the properties that define the clustering communication protocol stack.
  - `propertyString`: A string containing the properties that define the clustering communication protocol stack.
  - `url`: A link to the XML configuration file that contains the properties that define the clustering communication protocol stack.
- `protocol`: Defines the mechanism to use for data replication. Note that only one can be specified. The default protocol used is `multicast`.
  - `database`: The connection information required to persist state data to a database.
    - \* `dataSource`: The name of the data source that will provide the database connection. This must be the JNDI name for the data source, which is the value of the `jndi-name` attribute specified in `data-sources.xml`.
  - `peer`: Contains the configuration required to use peer-to-peer (P2P) communication for replication.
    - \* `range`: The number of times to increment the port value while looking for a potential peer node. The default is 5 times. Valid only for configuring static peer-to-peer replication in a standalone OC4J installation.
    - \* `startPort`: The initial port to attempt to allocate for usage by this cluster configuration for peer communication. The default is port 7800. 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 (3 seconds). Valid only for configuring static peer-to-peer replication in a standalone OC4J installation.
    - \* `nodes`: Contains the host name and port of a node to poll for the list of available OC4J servers. Multiple nodes may be specified. Valid only for configuring static peer-to-peer replication in a standalone OC4J installation.
    - \* `bindAddr`: Optionally specifies the IP address of a 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.
  - `multicast`: Contains the configuration required to use multicast communication for replication. This is the default mechanism used.
    - \* `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.
    - \* `bindAddr`: Optionally specifies the IP address of a 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.

- `replicationPolicy`: Defines when replication of `HttpSession` or stateful session bean state occurs, and whether all attributes/variable values or changed values only 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.
  - `scope`: Defines what data is replicated: Either all attribute or variable values, or changed values only. By default, only modified HTTP session attributes are replicated; for stateful session beans, all member variables are replicated.
  - `trigger`: 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.

### **connectorsPath**

Defines a plug-in connector deployed with the application.

- `path`: The name and path of the `oc4j-connectors.xml` file. If not specified, then the default path is `ORACLE_HOME/j2ee/home/connectors/rarName./oc4j-connectors.xml`.

### **dataSourcesPath**

Specifies the path and file name of the XML file defining data sources to be used by the application.

The default `data-sources.xml` file, which defines the default data source used by OC4J, is installed in the `ORACLE_HOME/j2ee/home/config/` directory.

- `path`: The path to the XML file. The path can be fixed or relative to the location of the `orion-application.xml` descriptor, which is `ORACLE_HOME/j2ee/home/config/` by default.

### **defaultDataSource**

Defines the default data source to use if other than server default. This must point to a valid CMT data source for this application if specified.

### **deploymentVersion**

Defines the version of OC4J that this JAR was deployed against, if it is not matching the current version then it will be redeployed. This is an internal server value; do not edit.

### **enableIIOP**

Set to `true` to enable IIOP. The default is `false`.

### **importedLibraries**

Defines one or more shared libraries to be imported by the application, as well as one or more shared libraries to delete from the set of libraries inherited by default from the parent application.

- `editImport`: Specifies a shared library to be imported by the application.
  - `maxVersion`: The highest implementation version of the shared library to import. (To import the latest version of a shared library, do not specify a version.)

- `minVersion`: The lowest implementation version of the shared library to import.
- `name`: The name of the shared library to import.
- `editRemove`: Specifies a shared library to be removed from the set of shared libraries inherited by default from the application's parent. This includes shared libraries inherited from the server-level `system` and `default` applications, which are inherited by all applications deployed into OC4J by default.
  - `name`: The name of the shared library to remove.

### **jazn**

Configures the Java Authentication and Authorization Service (JAAS) to use the XML-based configuration provider type.

- `provider`: Set to `XML`
- `location`: Set the path to the `jazn-data.xml` file. For example: `./jazn-data.xml`. This can be an absolute path, or a path relative to the `jazn.xml` file, where the JAAS Provider first looks for the `jazn-data.xml` in the directory containing the `jazn.xml` file. Optional if `jazn.xml` file configured, otherwise required.
- `persistence` Values are `NONE` (do not persist changes), `ALL` (persist changes after every modification), `VM_EXIT` (The default - persist changes when the JVM exits)
- `default-realm`: A realm name. For example: `sample_subrealm`. Optional if only one realm is configured.

### **jaznLoginConfig**

Contains data used to associate the application with a JAAS login module. These properties correspond to the `<jazn-loginconfig>` element and its sub-elements in the `jazn-data.xml` file, and should only be displayed if the JAZN XML-based provider will be used by the application.

- `className`: The login module implementation class.
- `controlFlag`: Indicates whether the login module is required to succeed to proceed with authentication. See the *Oracle Containers for J2EE Security Guide* for details.
- `options`: A collection of one or more properties to pass to the login module as name/value pairs. For example: name `debug`, value `true`.

### **jmxMBeans**

Defines one or more MBeans deployed with the application. Specify each property below for each MBean being deployed.

- `className`: The MBean implementation class.
- `description`: A string containing a readable name for the MBean. This name will be displayed in the Application Server Control Console user interface.
- `objectName`: The name to register the MBean under. The domain part of the name will be ignored even if specified; application MBeans are registered using the application's deployment name as the domain name.



For example, if you deploy an MBean named `MyMBeanA` with an application named `widget`, supply `supply:name=MyMBeanA` as the value of this attribute. The name will then be displayed as `widget:name=MyMBeanA`.

### libraries

Specifies either a relative or absolute path or URL to a directory or a JAR or ZIP archive to add as a library path for this OC4J instance. Directories are scanned for archives to include at OC4J startup.

- `path`: The path to the directory or archive.

### log

Sets the logging configuration for the application.

- `file`: The optional path to the directory where text log files will be generated if text logging will be used by the application. The default location for text log files is `ORACLE_HOME/j2ee/home/application-deployments/application_name/application.log`. Note that the default log file name and path should NOT be modified.
- `mail`: Optionally set the e-mail address to mail log output to. A valid mail-session must also be specified through the `mailSessions` property if this option is selected.
- `odl`: Configures Oracle Diagnostic Logging to be used by the application. The ODL framework 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 in XML, enabling them to be more easily parsed and reused by other Oracle Application Server and custom developed components.
  - `maxDirectorySize`: 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.
  - `maxFileSize`: The maximum size, in bytes, that an individual log file is allowed to grow to. When this limit is reached, a new log file is generated.
  - `path`: The path to the folder to output the `log.xml` files for this component to. The path can be absolute or relative to the XML configuration file you are modifying.

For example, to output `log.xml` files in a `/hello-planet-xml` directory within `ORACLE_HOME/j2ee/home/log/`, set the `path` attribute to `"../../log/hello-planet-xml"`.

### mailSessions

Defines the mail session SMTP host, if SMTP is used by the application.

- `location`: The location within the namespace to store the mail session in.
- `smtp-host`: The session SMTP server host.

### namespaceAccess

Sets the namespace or naming context security policy for RMI clients.

### parentApp

Specifies the name of the application that serves as the parent of this application. This value is set during deployment.

**persistencePath**

Defines the path to a directory where application state data should be stored across application restarts. The path can be absolute or relative to the application root.

- `path`: The path to the persistence directory.

**resourceProviders**

Defines a resource provider.

- `className`: The name of the resource provider class.
- `name`: The name used to identify the resource provider. This name will be used in finding the resource provider in the application's JNDI as `"java:comp/resource/name/"`.
- `description`: An optional description of the specific resource provider.
- `properties`: Set name/value property pairs to pass to the resource provider as parameters.

**seeParentDataSources**

Specifies whether the application should inherit the existing data sources defined for its parent application. The default is `false`.

**taskManagerInterval**

Defines the interval at which the task manager performs its duties, in milliseconds. The task manager is a background process that performs cleanup activities. By default, it is started every second (1000 milliseconds).

**treatZeroAsNull**

Specifies whether to read the value zero as null when it represents a primary key. The default is `false`.

**userManagerByClass**

Specifies an optional `UserManager` class to use, which can be any class that implements the `com.evermind.security.UserManager` interface. For example, `com.evermind.sql.DataSourceUserManager` or `com.evermind.ejb.EJBUserManager`. These classes are typically used to integrate existing systems and provide custom user-managers for Web applications.

- `className`: The fully qualified classname of the `UserManager`.
- `description`: An optional text description.
- `displayName`: A descriptive name for this `UserManager` instance.
- `properties`: One or more properties as name/value pairs to be passed to the `UserManager`. For example:

```
name="groupMembershipGroupName" value="group"  
name="groupMembershipUsernameFieldName" value="userid"
```

**webSiteBinding**

Specifies the name of the Web site the application's Web module is bound to. By default, all applications deployed into OC4J are bound to the `default` Web site. See the *Oracle Containers for J2EE Configuration and Administration Guide* for details on creating and using Web sites in OC4J.

## Setting Web Module Configuration Properties

The following details the OC4J-specific properties that can be set when deploying a Web module packaged in a WAR file. Each property maps to an element attribute in the `orion-web.xml` descriptor.

### **accessMask**

Specifies optional access masks for this application. You can specify host names or domains to filter clients through the `hostAccess` property; specify IP addresses and subnets to filter clients in `ipAccess`; or define both.

- `default`: Specifies whether to allow requests from clients not identified through a `hostAccess` or `ipAccess` property. Supported values are `allow` (default) and `deny`.
- `hostAccess`: Specifies a host name or domain from which to allow or deny access.
  - `domain`: The host or domain.
  - `mode`: Whether to allow or deny access from the specified host or domain. Supported values are `allow` (default) or `deny`.
- `ipAccess`: Specifies an IP address and subnet mask from which to allow or deny access.
  - `ip`: The IP address, as a 32-bit value; for example: `123.124.125.126`.
  - `netmask`: The relevant subnet mask; for example: `255.255.255.0`.
  - `mode`: Whether to allow or deny access from the specified IP address and subnet mask. Supported values are `allow` (default) or `deny`.

### **autoJoinSession**

Specifies whether users should be assigned a session as soon as they log in to the application. The default is `false`.

### **classpath**

Specifies additional code locations for Web application classloading. These can be either library files or locations for individual class files.

- `path`: The path(s) to one or more code locations, separated by commas or semicolons. A location can be one of the following:
  - The complete path to a JAR or ZIP file, including the file name
  - A directory path

If you specify a directory path, the classloader recognizes only individual class files in the specified directory, not JAR or ZIP files (unless those are specified separately).

The classloader recognizes the following:

- The `lib1.jar` and `zip1.jar` libraries (but no other libraries in `/abc/def`)
- Any class files in `/abc/def`
- Any class files in `mydir`, relative to the location of the generated `orion-web.xml` file

**defaultBufferSize**

Specifies the default size of the output buffer for servlet responses, in bytes. The default is 2048.

**defaultCharset**

Specifies the ISO character set to use by default. The default is "iso-8859-1".

**defaultMimeType**

Specifies the default MIME type to use for files with unknown or unrecognized extensions.

**deploymentVersion**

Specifies the version of OC4J under which this Web application was deployed. If this value does not match the current version, then the application is redeployed. This is an internal server value and should not be changed.

**development**

This property is a convenience for use during development. If set to `true`, then the OC4J server checks a particular directory for updates to servlet source files. If a source file has changed since the last request, then OC4J will, upon the next request, recompile the servlet, redeploy the Web application, and reload the servlet and any dependency classes.

The directory is determined by the setting of the `sourceDirectory` attribute (described below).

**directoryBrowsing**

Specifies whether to allow directory browsing for a URL that ends in `/`. Values are `allow` and `deny` (default).

Assume the following circumstances:

- There is no `index.html` file in the directory the URL is mapped to.
- There is no "welcome" page defined in the `web.xml` file.

If set to `allow` under these circumstances, then a URL ending in `/` results in the contents of the corresponding directory being displayed in the user's browser.

If set to `deny` under these circumstances, then a URL ending in `/` results in an error indicating that the directory contents cannot be displayed.

If there *is* a defined welcome file or there is an `index.html` file in the directory the URL is mapped to, then the contents of that file are displayed, regardless of this setting.

**enableJspDispatcherShortcut**

A `true` setting, which is the default, results in significant performance improvements by the OC4J JSP container, especially in conjunction with a `true` setting for the `simpleJspMapping` property. This is particularly true for JSP pages with numerous `jsp:include` tags.

Use of the `true` setting assumes, however, that if you define JSP files with `<jsp-file>` elements in `web.xml`, you have corresponding `url-pattern` definitions for those files.

**expirationSettings**

Defines the length of time before a specified set of resources - such as image files - will expire in the user's browser. This is useful for caching policies, such as for not reloading images as frequently as documents.

- `expires`: The number of seconds before the resources expiration, or "never" for no expiration. The default setting is "0" (zero), for immediate expiration.
- `urlPattern`: The URL pattern that the expiration applies to, such as in the following example:

```
urlPattern="*.gif"
```

**fileModificationCheckInterval**

Defines the interval, in milliseconds, at which OC4J checks for modified files. This property applies only to static files, such as HTML files.

Within the time period since the last check, further checks are not necessary. Zero or a negative number specifies that a check always occurs. The default is 1000. For performance reasons, a very large value (1000000, for example) is recommended in a production environment.

**id**

Defines a unique internal identifier generated at the time of deployment.

**jaznWebApp**

Configures the OracleAS JAAS Provider and Single Sign-On (SSO) properties for servlet execution. You must set these features appropriately to invoke a servlet under the privileges of a particular security subject.

- `authMethod`: Supported values are `BASIC` (for basic J2EE authentication, the default) and `SSO`. Use `SSO` to employ Oracle Application Server Single Sign-On for HTTP client authentication. Use `BASIC` mode if the application uses a custom `LoginModule` instance.
- `runAsMode`: Set to `true` to invoke the servlet using the privileges of a particular *subject*. A subject is defined by an instance of the `javax.security.auth.Subject` class and includes a set of facts regarding a single entity, such as a person. Such facts include identities and security-related attributes, such as passwords and cryptographic keys.

With the default `runas-mode="false"` setting, `doasprivileged-mode` is ignored.

- `doAsPrivilegedMode`: Assuming `runAsMode` is set to `true`, use the default `true` setting to use privileges of a particular subject without being limited by the access-control restrictions of the server.

For additional information about JAAS and the features described for this element, see the *Oracle Containers for J2EE Security Guide*.

**jspCacheDirectory**

Specifies the JSP cache directory, which is used as a base directory for output files from the JSP translator. It is also used as a base directory for application-level TLD caching. The default value is `./persistence`, relative to the application deployment directory.

**jspCacheTlds**

Indicates whether persistent TLD caching is enabled for JSP pages. TLD caching is implemented both at a global level, for TLD files in "well-known" tag library locations, and at an application level, for TLD files under the `/WEB-INF` directory. Values are `standard` (default), `on` or `off`. Well-known locations are defined in the `jspTaglibLocations` property (documented below). For more information on these values, see the *Oracle Containers for J2EE Support for JavaServer Pages Developer's Guide*.

**jspPrintNull**

Set to `false` to print an empty string instead of the default "null" string for null output from a JSP page. The default is `true`.

**jspTaglibLocations**

Defines the locations of shared tag libraries. This value can be set at the global application level only.

**jspTimeout**

Specifies an integer value, in seconds, after which any JSP page will be removed from memory if it has not been requested. This frees up resources in situations in which some pages are called infrequently. The default value is 0 (zero), for no timeout.

**mimeMappings**

Defines the path to a file containing MIME mappings to use.

- `path`: The path or URL for the file, either absolute or relative to the location of the `orion-web.xml` file.

**persistencePath**

Indicates where to store servlet `HttpSession` objects for persistence across server restarts or application redeployments. Specify a relative path, which will be relative to an OC4J temporary storage area under the `application-deployments` directory. There is no default value. If no value is defined, then there is no persistence of session objects across restarts or redeployments.

Session objects must be serializable (directly or indirectly implementing the `java.io.Serializable` interface) or remoteable (directly or indirectly implementing the `java.rmi.Remote` interface) for this feature to work.

**requestTrackers**

Specifies one or more servlets to use as dedicated request trackers. Request trackers are useful for logging information, for example.

A request tracker is invoked for each separate request sent from a browser to the server, at the time that the corresponding response is committed (immediately before the response is actually sent).

There can be multiple request trackers, each one defined in a separate property.

- `servletName`: The servlet to invoke. You can specify either the servlet name or the class name, according to the corresponding `<servlet-name>` or `<servlet-class>` element (both of which are subelements of a `<servlet>` element) in the `web.xml` file.

**servletChaining**

Specifies a servlet to call when the response of the current servlet is set to a specified MIME type. The specified servlet is called after the current servlet. This is known as *servlet chaining*, for filtering or transforming certain kinds of output.

- `mimeType`: The MIME type used to trigger the chaining, such as "text/html".
- `servletName`: The servlet to call when the specified MIME type is encountered. The servlet name is tied to a servlet class through its definition in the `<web-app>` element of `global-web-application.xml`, `web.xml`, or `orion-web.xml`.

**servletWebdir**

Specifies the path for invoking a servlet by class name. Anything appearing after this path in a URL is assumed to be a class name, including the package.

The following example illustrates servlet invocation by class name, assuming this property is set to `/servlet/`:

```
http://www.example.com:8888/servlet/foo.bar.SessionServlet
```

This feature is typically for use in an OC4J standalone environment during development and testing. For deployment, use the standard `web.xml` mechanisms for defining the context path and servlet path.

**sessionTracking**

Specifies the session-tracking settings for this application. Session tracking is accomplished using cookies, assuming a cookie-enabled browser.

- `cookies`: Set to `enabled` (default) to send session cookies; set to `disabled` to disable this setting.
- `cookieDomain`: Set to the desired domain to use for cookies. You can use this attribute to track a single client or user over multiple Web sites. The setting must start with a period (".") and must consist of at least two elements, such as `.us.oracle.com` or `.oracle.com`.

In this case, the same cookie is used when the user visits any site that matches the `.us.oracle.com` domain pattern, such as `webserv1.us.oracle.com` or `webserv2.us.oracle.com`.

- `cookieMaxAge`: Set the maximum length of time, in seconds, that the browser will keep a cookie. By default, the cookie is kept in memory during the browser session and discarded afterward.
- `sessionTracker`: The name of the servlet to use as a session tracker.
 

The session tracker is invoked as soon as a session is created; specifically, at the same time as the invocation of the `sessionCreated()` method of the HTTP session listener (an instance of a class implementing the `javax.servlet.http.HttpSessionListener` interface).
- `servletName`: The servlet to invoke as a session tracker. Specify either the servlet name or the class name, according to the corresponding `<servlet-name>` or `<servlet-class>` element, both of which are subelements of a `<servlet>` element) in the Web module's `web.xml` file.

**simpleJspMapping**

Set to `true` if `*.jsp` is mapped *only* to the OC4J front-end JSP servlet - `oracle.jsp.runtimev2.JspServlet` - in the `<servlet>` elements of any Web descriptors affecting your application (`global-web-application.xml`, `web.xml`,

and `orion-web.xml`). This allows performance improvements for JSP pages. The default setting is `false`.

### **sourceDirectory**

Specifies the location of servlet source files to auto-compile if the `development` property is set to `true`. The default location is `/WEB-INF/src` if it exists; otherwise it is `/WEB-INF/classes`.

### **temporaryDirectory**

Contains the path to a temporary directory that can be used by servlets and JSP pages for scratch files. The path can be either absolute or relative to the deployment directory. The default setting is `./temp`.

A servlet may use a temporary directory, for example, to write information to disk as a user is entering data in a form, for interim or short-term storage before the information is written to a database.

### **virtualDirectories**

Adds a virtual directory mapping for static content. This is conceptually similar to symbolic links on a UNIX system, for example.

The virtual directory enables you to make files in the "real" document root directory available to the application, even though the files do not physically reside in the Web application WAR file. This would be useful, for example, to link an enterprise-wide error page into multiple WAR files.

- `realPath`: The real path to the files that the virtual path will map to. For example, `/usr/local/realpath` in UNIX or `C:\testdir` in Windows.
- `virtualPath`: The virtual path to map to the specified real path.

### **webAppClassLoader**

Contains class loading instructions passed to the OC4J server at Web module startup.

- `searchLocalClassesFirst`: Set to `true` to search and load WAR file classes before system classes. The default setting is `false`.
- `includeWarManifestClassPath`: Set to `false` to *not* include the classpath specified in the WAR file manifest `Class-Path` attribute when searching and loading classes from the WAR file, regardless of the `searchLocalClassesFirst` setting. The default setting is `true`.

If both attributes are set to `true`, the overall classpath is constructed so that classes physically residing in the WAR file are loaded prior to any classes defined in the WAR file manifest classpath. In the event of conflict, classes physically residing in the WAR file will take precedence.

## **Setting Enterprise JavaBeans Module Configuration Properties**

The following details the OC4J-specific properties that can be set when you are deploying a J2EE application packaged in an EAR file. Each property maps to an element attribute in the `orion-ejb.xml` descriptor.

The properties that can be set differ for entity beans, session beans and message-driven beans. As such, the properties are broken out as follows:

- [General EJB Properties](#)
- [Setting Entity Bean Properties](#)



- [Setting Session Bean Properties](#)
- [Setting Message-Driven Bean Properties](#)

### General EJB Properties

The following properties apply to all EJBs within the archive being deployed.

#### **defaultMethodAccess**

Sets the default method access policy for insecure methods not associated with a role mapping. Methods are automatically mapped to the default security role.

- `impliesAll`: Set to `true` (default) to disable security role checking for insecure methods.  
If `false`, you must map the default role defined in the `name` attribute to an OracleAS JAAS Provider or XML user or group through the `user` and `group` properties (described below). The default is `false`.
- `name`: The default security role that insecure methods will be mapped to. The default is the `<default-ejb-caller-role>` role; however, this value can be changed to any valid role.
- `groups`: One or more group names that will be used by clients to access insecure methods.
- `users`: One or more user names that will be used by clients to access insecure methods.

#### **deploymentVersion**

The version of OC4J the EJB JAR is being deployed into. This is an internal server value.

#### **persistenceManager**

Defines the persistence manager component to use to manage the persistence layer of entity EJBs.

The TopLink utility is the default persistence manager used with OC4J, and by default, all EJBs deployed into OC4J are managed by the TopLink PM. No configuration is required for TopLink. See the *Oracle TopLink Getting Started Guide* for details.

- `descriptor`: The file name of the persistence manager's deployment descriptor file.
- `name`: The name of the persistence manager implementation to use. Valid values are `toplink` if using TopLink, or `orion` if using the Orion CMP implementation. The default is `toplink`.
- `pmProperties`: Contains configuration properties for the TopLink persistence manager.
  - `customizationClass`: An optional Java class implementing the `oracle.toplink.ejb.cmp.DeploymentCustomization` interface used to allow deployment customization of TopLink mapping and runtime configuration. The class must be fully qualified and included in the EJB JAR being deployed.
  - `dbPlatformClass`: A TopLink database platform class containing TopLink support specific to a particular database. The specified class must be fully qualified.

- `projectClass`: An optional TopLink project class containing mapping metadata. This class will replace the TopLink descriptor specified in the `descriptor` property. The class must be fully qualified and must exist in the EJB JAR file being deployed.
- `remoteRelationships`: Set to `true` to maintain relationships between remote objects through the entities' remote interfaces. Note that this flag is not compliant with EJB 2.0. The default is `false`.
- `sessionName`: A unique name for the EJB JAR being deployed. The name must be unique among all TopLink-persisted JARs deployed in OC4J. If no value is specified, a unique name will be generated by the TopLink persistence manager.
- `mode`: Set whether updates should be propagated to another data store server synchronously or asynchronously. The default value is `asynchronously`.
- `serverUrl`: The URL to the data store host.
- `serverUser`: The user name to use to access the host.
- `dbTableGen`: Specifies how TopLink will create or use the database tables being mapped to. This setting is ignored if the mappings are already defined for the entities. Values are:
  - \* `Create`: Attempt to create the tables. This is the default.
  - \* `DropAndCreate`: Attempt to drop existing tables before re-creating them.
  - \* `UseExisting`: Use existing tables.
- `extendedTableNames`: Set to `true` only if the generated table names are not long enough to be unique. This setting is ignored if the mappings are already defined for the entities. The default is `false`.

### Setting Entity Bean Properties

The following properties apply to an entity bean included in the EJB archive.

#### **callTimeout**

Specifies the maximum time, in milliseconds, to wait for any resource to make a business/life-cycle method invocation. This is not a timeout for how long a business method invocation can take. The default is 90000 milliseconds (90 seconds).

If the timeout is reached, a `TimedOutException` is thrown. This excludes database connections.

Set to 0 if you want the timeout to be forever. See the EJB section in the *Oracle Application Server Performance Guide* for more information.

#### **cmpFieldMappings**

Defines information for a single container-managed persistence field. If no subtags are used to define different behavior, the field is persisted through serialization or native handling of "recognized" primitive types.

Attributes:

- `name`: The name of the persistence field.
- `ejbReferenceHome`: The JNDI location of the persistence fields. This will be the remote EJB home if the field is an entity `EJBObject` or an `EJBHome`.
- `persistenceName`: The name of the field in the database table.

- `persistenceType`: The field's type. Values are database-specific.

**copyByValue**

Specifies whether or not to copy (clone) all of the incoming and outgoing parameters in EJB calls. Set to `false` if you are certain that your application does not assume copy-by-value semantics for a speed-up. The default is `true`.

**dataSource**

Specifies the name of the data source used if using container-managed persistence.

**delayUpdatesUntilCommit**

Specifies when to flush cached data. If `true`, the flushing of transactional data is deferred until database commit time. Set to `false` to update persistence data after completion of every EJB method invocation, except `ejbRemove()` and finder methods. The default is `true`.

**disableWrapperCache**

Set to `true` to disable the wrapper class cache for the EJB.

**doSelectBeforeInsert**

Specifies whether to execute a `SELECT` statement before an insert into the database. The extra select normally checks to see if the entity already exists before doing the insert to avoid duplicates.

The default value is `true`. For performance, Oracle recommends setting this property to `false` to avoid executing the extra statement.

However, if a unique key constraint is not defined for the entity, setting this value to `false` will allow a duplicate insert to avoid detection. To prevent duplicate inserts in this case, leave the value set to `true`.

**exclusiveWriteAccess**

Specifies whether the EJB server has exclusive write (update) access to the database. This can be used only for entity beans that use a "read only" locking mode. In this case, it increases the performance for common bean operations and enables better caching.

This parameter corresponds to the commit option used (A, B or C, as defined in the EJB specification). When this property is set to `true`, commit option A is used.

Default is `false` for beans where `lockingMode` is set to `optimistic` or `pessimistic`, or `true` if `lockingMode` is set to `read-only`.

**findByPrimaryKeyLazyLoading**

Set to `true` to turn on lazy loading and enforce only a single execution of the `select()` finder method. For entity bean finder methods, lazy loading can cause this method to be invoked more than once. The default is `false`.

**finderMethods**

Contains properties defining the selection criteria in a `findByXXX()` method in the EJB's home interface.

- `partial`: Specifies whether or not the specified query is a partial query. A partial query is the 'where' clause or the 'order' (if it starts with `order`) clause of the SQL query. The default is `true`, indicating queries are partial by default.

If set to `false`, then the full query is to be entered as the value for the query attribute. Ensure that the query produces a result-set containing all of the CMP fields. This is useful when doing advanced queries involving table joins and similar operations.

- `query`: Defines the query part of an SQL statement, which is the section following the `WHERE` keyword. Special tokens are `$number`, which denotes a method argument number; and `$name`, which denotes a cmp-field name.

For example, assuming the cmp-field is named `age`, the query for `findByAge(int age)` would be `$1 = $age`.

- `lazyLoading`: Set to `true` to turn on lazy loading and enforce only a single execution of the `select()` finder method. The default is `false`.
- `prefetchSize`: Oracle JDBC drivers include extensions that allow you to set the number of rows to prefetch into the client while a result set is being populated during a query. This reduces round trips to the database by fetching multiple rows of data each time data is fetched--the extra data is stored in client-side buffers for later access by the client. The number of rows to prefetch can be set as desired.

The default number of rows to prefetch to the client is 10. See the *Oracle Database JDBC Developer's Guide and Reference* for more information on using prefetch with a JDBC driver.

### **forceUpdate**

Set to `true` to allow OC4J to execute the EJB lifecycle by invoking the `ejbStore()` method. This manages data in transient fields and sets appropriate persistent fields during the method call. For example, an image might be kept in one format in memory, but stored in a different format in the database. The default is `false`.

### **instanceCacheTimeout**

Specifies the amount of time, in seconds, that entity wrapper instances are assigned to an identity. Set to `never` to retain the wrapper instances until they are garbage collected. The default is 60 seconds.

### **iorSecurityConfigs**

Configures CSIV2 security policies for interoperability. See *Oracle Containers for J2EE Security Guide* for details.

### **isolation**

Specifies the isolation level for database actions.

- Valid values for Oracle databases are `serializable` and `committed` (default).
- Valid values for non-Oracle databases include the following: `none`, `committed`, `serializable`, `uncommitted`, and `repeatable_read`.

### **location**

Defines the JNDI name this EJB will be bound to.

### **lockingMode**

Configures the concurrency modes, which specify when to block for resource contention management, or when to execute in parallel. Values are:

- **optimistic**: Allows multiple users to execute the entity bean in parallel. It does not monitor resource contention; thus, the burden of the data consistency is placed on the database isolation modes. This is the default.
- **pessimistic**: Manages resource contention and does not allow parallel execution. Only one user at a time is allowed to execute the entity bean at a single time.
- **read-only**: Allows multiple users to execute the entity bean in parallel. However, the container does not allow any updates to the bean's state.

**maxInstances**

Sets the number of maximum bean implementation instances to be kept instantiated or pooled. The default is 100. Set to 0 to indicate no maximum.

**maxTxRetries**

Specifies the number of times to retry a transaction that was rolled back due to system-level failures. The default is 3.

Generally, you should add retries only where errors are seen that could be resolved through retries. For example, if you are using serializable isolation and you want to retry the transaction automatically if there is a conflict, you might want to use retries.

**minInstances**

Sets the number of minimum bean implementation instances to be kept instantiated or pooled. The default is 0.

**name**

Specifies the name of the bean, which matches the name specified in the assembly section of the standard EJB deployment descriptor (`ejb-jar.xml`).

**poolCacheTimeout**

Defines the amount of time, in seconds, that the bean implementation instances are to be kept in the "pooled" or unassigned state. Set to 0 to retain bean instances until they are garbage collected. The default is 60.

**primkeyMapping**

Maps the persistent field that serves as the EJB primary key to the corresponding column in the database table where data will be persisted.

Attributes:

- **name**: The EJB persistent field.
- **persistenceName**: The database column.

**table**

Contains the name of the database table where data will be persisted if using container-managed persistence.

**txRetryWait**

Specifies the time to wait in seconds between retrying the transaction. The default is 60 seconds.

**updateChangedFieldsOnly**

Specifies whether the container updates either modified fields only or all fields to persistence storage for CMP entity beans when `ejbStore()` is invoked. The default is `true`, which specifies that only modified fields are updated.

**validityTimeout**

Sets the maximum amount of time, in milliseconds, that an entity is valid in the cache before being reloaded. This is useful for loosely coupled environments where rare updates from legacy systems occur. This attribute is only valid for entity beans with a locking mode of `read_only` and the `exclusiveWriteAccess` property set to `true` (the default).

If the EJB is generally not modified externally, meaning that the table is occasionally updated and cache updates are required, set this to a value corresponding to the interval at which you think the data may be changing externally.

If the data is never modified externally the value can be set to `0` or `-1` to disable this option. The data in the cache will always be valid for read-only EJBs that are never modified externally.

**Setting Session Bean Properties**

The following properties apply to an entity bean included in the EJB archive.

**callTimeout**

Specifies the maximum time, in milliseconds, to wait for any resource to make a business/life-cycle method invocation. This is not a timeout for how long a business method invocation can take. The default is `90000` milliseconds (90 seconds).

Set to `0` for no timeout. See the EJB section in the Oracle Application Server 10g Performance Guide for more information.

**copyByValue**

Specifies whether or not to copy (clone) all of the incoming and outgoing parameters in EJB calls. Set to `false` if you are certain that your application does not assume copy-by-value semantics for a speed-up. The default is `true`.

**idleTime**

Specifies the timeout, in seconds, applied to stateful session EJBs. If the value is `0` or negative, then all timeouts are disabled. The default is `1800` seconds (30 minutes).

The timeout parameter is an inactivity timeout for stateful session beans. Every 30 seconds the pool clean up logic is invoked. Within the pool clean up logic, only the sessions that timed out, by passing the timeout value, are deleted.

Adjust the timeout based on your application's use of stateful session beans. For example, if stateful session beans are not removed explicitly by your application, and the application creates many stateful session beans, then you may want to lower the timeout value.

**iorSecurityConfigs**

Configures CSIv2 security policies for interoperability. See the *Oracle Containers for J2EE Security Guide* for details.

**location**

Defines the JNDI name this EJB will be bound to.

**maxInstances**

Defines the maximum number of bean instances, either instantiated or pooled, allowed in memory. The default is 100.

When this limit is reached, the container attempts to passivate the oldest bean instance from memory. If unsuccessful, the container waits the number of milliseconds set in `callTimeout` to see if a bean instance is removed from memory, through passivation, its `remove()` method, or bean expiration, before a `TimeoutExpiredException` is thrown back to the client.

Set to 0 to allow an infinite number of bean instances. This property applies to both stateless and stateful session beans.

**maxInstancesThreshold**

Specifies how many active beans can exist in relation to the value set for `maxInstances` before passivation is initiated.

Specify an integer that is translated as a percentage. For example, if `maxInstances` is 100 and `maxInstancesThreshold` is 90 percent, passivation of beans occurs when active bean instances reaches past a total of 90.

The default is 90. Set to 0 to disable this feature.

**maxTxRetries**

Specifies the number of times to retry a transaction that was rolled back due to system-level failures. The default is 3.

Generally, you should add retries only where errors are seen that could be resolved through retries. For example, if you are using serializable isolation and you want to retry the transaction automatically if there is a conflict, you might want to use retries.

**memoryThreshold**

Defines a threshold for how much used JVM memory is allowed before passivation should occur. Specify an integer that is translated as a percentage. When the percentage is reached, beans are passivated, even if their idle timeout has not expired. The default is 80 percent. To disable, specify `never`.

**minInstances**

Sets the number of minimum bean implementation instances to be kept instantiated or pooled. The default is 0.

**name**

Defines the name of the EJB, which matches the name of a bean in the assembly section of the standard EJB deployment descriptor (`ejb-jar.xml`).

**passivateCount**

Defines the number of beans to be passivated if any of the resource thresholds - such as `maxInstancesThreshold` or `memoryThreshold` - has been reached. Passivation of beans is performed using the least recently used algorithm. By default, the number of beans is one-third of the value set for `maxInstances`. Set the count to 0 or a negative number to disable.

**persistenceFilename**

Defines the path to the file where sessions are stored across OC4J restarts.

**poolCacheTimeout**

Sets the amount of time, in seconds, that stateless sessions are to remain cached in the pool. At the specified interval, all unassigned beans in the pool are removed. The default is 60 seconds.

If the value specified is 0 or negative, beans are not removed from the pool.

**replication**

Defines when state replication should occur for stateful session beans in a clustered environment. Values are `VMTermination`, `EndOfCall` or `None` (default). See the *Oracle Containers for J2EE Configuration and Administration Guide* for details.

**resourceCheckInterval**

Defines the interval, in seconds, at which OC4J checks all resources to see which have passed the specified thresholds, such as `maxInstancesThreshold`. Passivation of beans occurs if any thresholds have been reached. The default is 180 seconds (3 minutes). To disable, set to 0.

**timeout**

Defines the inactivity timeout, in seconds, after which stateful session beans are considered ready for deletion from the pool. If the value is 0 or negative, then all timeouts are disabled. The default is 1800 seconds (30 minutes).

Adjust the timeout based on your application's use of stateful session beans. For example, if stateful session beans are not removed explicitly by your application, and the application creates many stateful session beans, then you may want to lower the timeout value. Note that the pool clean-up logic is invoked every 30 seconds.

**txRetryWait**

Specifies the length of time to wait, in seconds, between retrying a transaction. The default is 60 seconds.

**Setting Message-Driven Bean Properties**

The following properties apply to a message-driven bean within the archive being deployed. Each property pertains to attributes or sub-elements of a `<message-driven-deployment>` element in the `orion-ejb-jar.xml` descriptor.

**connectionFactoryLocation**

Contains the JNDI location of the connection factory to use.

The syntax is `java:comp/resource + resource_provider_name + TopicConnectionFactory` or `QueueConnectionFactory + user_defined_name`. The `ConnectionFactory` property details the type of factory that is being defined.

**dequeueRetryCnt**

Specifies how often the listener thread tries to re-acquire the JMS session once database failover has occurred. The default is 0. This value is only for container-managed transactions.

**dequeueRetryInterval**

Specifies the interval between attempts to re-acquire the JMS session. The default is 60 seconds.



**destinationLocation**

Specifies the JNDI location of the destination (queue/topic) to use.

The syntax is `java:comp/resource + resource_provider_name + Topics or Queues + Destination name`. The `Topic` or `Queue` details what type of Destination is being defined. `Destination name` is the actual queue or topic name defined in the database.

**listenerThreads**

Sets the number of listener threads spawned to listen for incoming JMS messages on the topic or queue. The threads concurrently consume JMS messages. Topics can only have one thread; queues can have more than one thread. The default is 1 thread.

**maxInstances**

Do not set this property; use `listenerThreads` instead.

**minInstances**

Do not set this property.

**name**

Contains the name of the EJB, which matches the name of an EJB in the assembly section of the J2EE standard EJB deployment descriptor (`ejb-jar.xml`) packaged in the EJB archive.

**resourceAdapter**

Contains the name of the resource adapter to be created by the connection factory for use by this bean. This value is defined in `orion-ra.xml`, the OC4J-specific resource adapter descriptor.

**subscriptionName**

Contains the topic subscription name, if the bean represents a topic.

**transactionTimeout**

Sets the transaction timeout interval, in seconds, for any container-managed transactional MDB. The default is 86400 seconds (one day). If the transaction has not completed within this timeframe, the transaction is rolled back.

## Setting Web Services Configuration Properties

Each property maps to an element attribute in the `oracle-webservices.xml` descriptor.

- [Setting General Web Services Properties](#)
- [Setting Web Service Description Properties](#)

### Setting General Web Services Properties

The following properties apply to all Web Services deployed with the archive.

**contextRoot**

Specifies the root context of the exposed Web Service, which is required only for EJB 2.1 Web Services. If the root context is not specified, the server defaults to the EJB JAR

file name minus the `.jar` extension. For example, `foo-ejb.jar` will be translated into `/foo-ejb`.

For Java class Web Services, the context root is derived from the context root specified in the parent application's `application.xml` descriptor.

### Setting Web Service Description Properties

The following properties apply to a Web Service contained within the archive being deployed.

#### **downloadExternalImports**

Specifies whether relative imports should be downloaded and resolved to absolute URLs. If set to `true`, `resolveRelativeImports` is automatically set to `true`. The default is `false`.

#### **exposeTestpage**

Set to `true` to expose the test page. The default is `true`.

#### **exposeWsdI**

Set to `true` to expose the WSDL file describing the Web service. The default is `true`.

#### **name**

Defines the name of the Web service for an EJB implemented as a Web service. This value matches the name defined in the J2EE standard EJB deployment descriptor (`ejb-jar.xml`).

#### **resolveRelativeImports**

Set to `true` to force relative imports to be resolved to absolute URLs. This property is automatically set to `true` if `downloadExternalImports` is `true`.

#### **wsdlFileFinalLocation**

Specifies the URI for the final updated WSDL that describes the Web service.

#### **wsdlPublishLocation**

Specifies the URI to the WSDL if `exposeWsdI` is set to `true`.

## Setting Application Client Configuration Properties

The following details the OC4J-specific properties that can be set when deploying an application client packaged in a JAR or CAR file. Each property maps to an element attribute in the `orion-application-client.xml` descriptor.

### **clientInvocationMappings**

Configures the client module to start when the OC4J container is started.

- `autoStart`: Set to `true` to start the client when OC4J is started. The default is `false`.
- `user`: The unique identifier of the user the client will run as.
- `path`: The path to the client JAR file. May be absolute or relative to the parent application's root.
- `arguments`: One or more string arguments to be passed to the client's `main()` method at startup.

**mailSessions**

Contains the configuration for a mail session resource.

- `description`: An optional description for the resource.
- `location`: The JNDI name to bind the mail session to.
- `smtpHost`: The name or IP address of the SMTP host to use, if using SMTP.
- `username`: A username used to log in to the resource, if required.
- `password`: The password used to access the resource.
- `properties`: Can contain additional properties that should be given to the mail-session as name/value pairs. For example:

```
name="mail.from" value="mail.sender@server.com
name="mail.transport.protocol" value="smtp"
name="mail.smtp.from" value="mail.sender@server.com
```

**ejb-ref**

Declares a reference to the home interface of an EJB that will be used by the client.

- `location`: The JNDI location the EJB is bound to.
- `name`: The EJB reference name used by the application client.

**Setting Resource Adapter Properties**

The following details the OC4J-specific properties that can be set when deploying a resource adapter packaged in a RAR file. Each property maps to an element attribute in the `oc4j-ra.xml` descriptor.

See the *Oracle Containers for J2EE Resource Adapter Administrator's Guide* for detailed information on resource adapter configuration.

**connectionPools**

Contains a name and one or more name/value pairs defining a shared connection pool. Valid property names are as follows:

- `maxConnections`: The maximum number of connections permitted within the pool. If no value is specified, there is no limit on the number of connections.
- `minConnections`: The minimum number of connections. If greater than 0, the specified number of connections are opened when OC4J is initialized. OC4J may not be able to open the connections if necessary information is unavailable at initialization time.

For instance, if the connection requires a JNDI lookup, it cannot be created, because JNDI information is not available until initialization is complete. The default value is 0.

- `scheme`: Defines how OC4J handles connection requests after the maximum permitted number of connections is reached. One of the following values must be specified:
  - `dynamic`: OC4J creates a new connection and returns it to the application, even if this violates the maximum limit. When these limit-violating connections are closed, they are destroyed instead of being returned to the connection pool.

- `fixed`: OC4J raises an exception when the application requests a connection and the maximum limit has been reached.
- `fixed_wait`: OC4J blocks the application's connection request until an in-use connection is returned to the pool. If `waitTimeout` is specified, OC4J throws an exception if no connection becomes available within the specified time limit.
- `waitTimeout`: The maximum number of seconds that OC4J waits for an available connection if `maxConnections` has been exceeded and the `fixed_wait` scheme is in effect. In all other cases, this property is ignored.

### **connectorFactories**

Contains properties defining an installed J2EE Connector Architecture-compliant resource adapter.

- `connectionFactoryInterface`: The name of the factory that will create managed connection instances, such as `javax.jms.QueueConnectionFactory`.
- `connectorName`: The name of the connector.
- `description`: An optional short description of the connector.
- `location`: The JNDI location which OC4J will bind the connection factory to. For example, if the value is set to `eis/myEIS1`, an application component can look up the connection factory using JNDI lookup on location `java:comp/env/eis/myEIS1`.
- `configProperties`: Configuration properties for the connector factory defined in `connectionFactoryInterface`.

Configuration properties are connector-specific and are defined in the `<config-property>` element in the J2EE deployment descriptor (`ra.xml`) packaged with the resource adapter. For a JCA 1.0 resource adapter, the `<config-property>` element is a sub-element of the `<resourceadapter>` element. For JCA 1.5, they are found inside the `<connection-definition>` element(s).

- `connectionPooling`: Contains properties defining a connection pool to use. Properties are:
  - `use`: Specifies whether a connection pool should be used for the connection factory; and if so, whether a shared pool or a private pool should be used. Possible string values for this attribute are:
    - \* `shared`: The shared connection pool to be used for this connection factory configuration.
    - \* `private`: The private, non-shared connection pool defined in the `properties` fields below will be used for this connection factory configuration.
    - \* `none`: Connection pooling is disabled for this connection factory configuration.
- `log`: The path to the log file generated by OC4J for this connector. If the path name is not specified or if the directory does not exist, logging is not enabled. For example: `./logConnFctry1.log`
- `securityConfig`: Specifies the security mechanism to use and corresponding credential information needed by the connector to access the EIS.

- 
- `use`: The security mechanism to use. Only one may be specified. Valid values are: `jaasModule`, `principalMappingEntries` or `principalMappingInterface`.
  - `jaasModule`: If using JAAS, the name of the JAAS application to use.
  - `principalMappingEntries`: If using the principal mapping entries mechanism, specify the following:
    - \* `defaultMapping`: Specify the resource username and password for the default resource principal. This principal is used to log on to the EIS if there is no principal mapping entry whose initiating user corresponds to the current initiating principal.
    - \* `principalMappingEntries`: Specify the initiating principal username and the resource username and password. This maps the initiating principal to the resource principal and password it will use to log in to the EIS.
  - `principalMappingInterface`: If using the OC4J-specific programmatic container-managed sign-on mechanism, which utilizes a class implementing the `oracle.j2ee.connector.PrincipalMapping` interface, specify the following:
    - \* `implClass`: The class implementing the `oracle.j2ee.connector.PrincipalMapping` interface. A JAR file containing the class must be placed into the directory containing the decompressed RAR file.
    - \* `properties`: The username and password that will be used to log in.
  - `xaRecoveryConfig`: The username and password of the privileged user that will log into the EIS during a two-phase commit transaction recovery.



---

---

# Deploying with Application Server Control Console

The *Oracle Enterprise Manager 10g Application Server Control Console* provides a Web-based interface for completing a range of deployment-related tasks, including:

- Deploying, undeploying and redeploying standalone modules (WARs, CARs, etc.) as well as complete J2EE applications into OC4J
- Creating and editing re-usable deployment plans
- Configuring application security and clustering at deployment time
- Importing shared libraries for use by the application, as well as modifying the list of inherited libraries
- Deploying an application to specific OC4J instances or to a "group" of instances within an Oracle Application Server cluster

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

This chapter includes the following sections:

- [Accessing Application Server Control Console](#)
- [Deploying to OC4J Instances Within a Cluster](#)
- [Deploying/Redeploying an Application](#)
- [Completing the Deployment Tasks](#)
- [Undeploying an Application](#)

## Accessing Application Server Control Console

The Application Server Control Console is installed and configured automatically when you install OC4J. This section covers the following topics:

- [Accessing Application Server Control Console in Standalone OC4J](#)
- [Accessing Application Server Control Console in Oracle Application Server](#)

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
```

## Deploying to OC4J Instances Within a Cluster

Application Server Control Console enables you to deploy an application to a specific OC4J instance or to a "group" of instances within an Oracle Application Server cluster. A *group* is defined as a loosely synchronized 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 an application can be deployed.



1. Click the **Cluster Topology** link on the Application Server Control Console home page. The resulting page displays the following:
  - All Oracle Application Server instances that are currently part of the cluster
  - The active OC4J instances within each Oracle Application Server instance
  - The applications deployed into each OC4J instance
2. Select the deployment target:
  - To deploy to a *group* of OC4J instances, click the common instance name shared by the group under Groups at the bottom of the page.
  - To deploy to a specific OC4J instance, click the link for the target instance you want to deploy the application to.
3. Once the target instance has been accessed, deploy the application by following the instructions in "[Deploying/Redeploying an Application](#)" on page 9-3.

## Deploying/Redeploying an Application

The Application Server Control Console provides a three-page deployment wizard that provides a streamlined, user-friendly deployment process.

---



---

**Note:** If the HTTP session times out due to browser inactivity while you are using the deployment wizard, you will have to re-start the deployment process from the beginning.

---



---

1. Click the **Applications** tab ->**Deploy** button to access the deployment wizard.
2. Select the archive to upload to the OC4J server in the first page of the wizard.
 

You also have the option to navigate to the location of an existing deployment plan, which can be applied to the archive or used as a template for a new deployment plan. If no deployment plan is specified, a new deployment plan will be created by default. See [Chapter 8, "Working With Deployment Plans"](#) for more information.
3. Set application attributes and bind the Web application to a Web site in the second page of the deployment wizard.

Specify the application or module name, which will be used to identify the application within OC4J. This name will also be displayed in Application Server Control Console.

Next, select the parent application of the application or module being deployed. If none specified, the `default` application is used.

Finally, if deploying a Web application, bind the application to the Web site that will be used to access it. This binding is accomplished by selecting the name of the XML configuration file that defines the Web site from the list.

The list contains all Web sites currently defined for the OC4J server instance. In most cases, applications will be bound to the `default` Web site, which is defined by the `default-web-site.xml` configuration file.

4. Complete the deployment tasks and/or edit the deployment plan directly before deploying the archive in the third page of the deployment wizard.

In this final screen, you have the option of completing a number of configuration tasks before deploying the application. These tasks provide an alternative to

editing the deployment plan, which contains configuration data that will be set in the OC4J deployment descriptors generated during deployment. See "[Completing the Deployment Tasks](#)" on page 9-4 for details on each optional task.

You also have the option of editing the deployment plan directly before deploying the archive. The edited deployment plan can then be saved for re-use. See [Chapter 8, "Working With Deployment Plans"](#) for more details on creating and editing deployment plans.

5. Deploy the application.

Note that the archive is not actually deployed until you click the **Deploy** button. The deployment plan, which up until this point exists only on the client side, will also be sent to the OC4J server with the archive. Note that once started, the deployment process will continue, even if the Web browser is closed.

## Completing the Deployment Tasks

The third panel in the Application Server Control Console deployment wizard gives you the option to complete a number of configuration tasks before deploying an application. Provide a more straightforward option for completing similar tasks through the deployment plan editor.

Just as with the deployment plan editor, values set through the various deployment tasks pages are written to the appropriate OC4J-specific deployment descriptor at deployment time.

## Selecting the Security Provider

OC4J supports two different provider types: XML for application development mode, and LDAP for production environments. Each provider type implements a repository for secure, centralized storage, retrieval, and administration of provider data.

- Select **File-Based Security Provider** to use the XML-based provider.

The XML-based provider is a lightweight implementation suitable for application prototyping in a development environment. User, realm, and policy information is stored in an XML file, normally `jazn-data.xml`.

- Select **Oracle Identity Management Security Provider** to use the LDAP-based Oracle Internet Directory provider.

This security provider is useful for applications being deployed into a production environment. User, realm, and policy information is persisted to the LDAP-based Oracle Internet Directory (OID).

Note that the OC4J instance must be configured to use Oracle Internet Directory before an application can be configured to use it.

- Select **Oracle Security Provider for 3rd Party LDAP Server** to use a non-Oracle LDAP provider.

Select this option to configure the application to use Active Directory, Sun Directory Server or another LDAP server. Use the tools provided by the LDAP server vendor for realm and principals management.

## Mapping Security Roles to Users and Groups

Map any security roles defined in your application to existing users and groups. If you have defined a security role within your application, you can map this role to a security group or role. You do not define security groups and users in this screen.

## Configuring Enterprise JavaBeans Included in the Application

You can configure a number of properties for the EJBs packaged with the application being deployed.

- Configure the following properties for each entity bean packaged with the application. The values displayed are the default values.

**Table 9–1 Entity Bean Properties**

Property	Values
Persistence Type	Indicates whether the bean will use bean-managed or container-managed persistence.
Min Instances	Sets the number of minimum bean implementation instances to be kept instantiated or pooled.
Max Instances	Defines the maximum number of bean instances, either instantiated or pooled, allowed in memory.
Max Transaction Retries	Specifies the number of times to retry a transaction that was rolled back due to system-level failures.
Pool Cache Timeout	Sets the amount of time, in seconds, that stateless sessions are to remain cached in the pool. At the specified interval, all unassigned beans in the pool are removed.
JNDI Name	Defines the JNDI name this EJB will be bound to.

- Select the data sources to associate with EJB modules containing entity beans.  
By default, all entity beans deployed into OC4J will use the Oracle TopLink persistence manager. Data sources can be created through the **Administration** tab -> **Services** -> **JDBC Resources** page in Application Server Control Console.
- Configure the following for each session bean packaged with the application. The values displayed are the default values.

**Table 9–2 Session Bean Properties**

Property	Values
Min Instances	Sets the number of minimum bean implementation instances to be kept instantiated or pooled.
Max Instances	Defines the maximum number of bean instances, either instantiated or pooled, allowed in memory.
Max Transaction Retries	Specifies the number of times to retry a transaction that was rolled back due to system-level failures.
Call Timeout	Specifies the maximum time, in milliseconds, to wait for any resource to make a business/life-cycle method invocation.
Pool Cache Timeout	Sets the amount of time, in seconds, that stateless sessions are to remain cached in the pool. At the specified interval, all unassigned beans in the pool are removed.
JNDI Name	Defines the JNDI name this EJB will be bound to.

- Configure the following for each message-driven bean packaged with the application. The values displayed are the default values.

**Table 9–3 Message-Driven Bean Properties**

Property	Values
Dequeue Retry Count	Specifies how often the listener thread tries to re-acquire the JMS session once database failover has occurred. Valid for container-managed transactions only.
Dequeue Retry Interval	Specifies the interval between attempts to re-acquire the JMS session.
Transaction Timeout	Sets the transaction timeout interval, in seconds, for any container-managed transactional MDB.
Number of Listener Threads	Sets the number of listener threads spawned to listen for incoming JMS messages on the topic or queue. Topics can only have one thread; queues can have more than one thread.

## Configuring Class Loading/Shared Libraries

Optionally manage the libraries imported by the application. By default, an application inherits the same set of shared libraries present in its parent application, including any shared libraries inherited from the `default` application.

The default page displays all of the shared libraries currently installed on the OC4J server instance. Shared libraries must have already been installed on the OC4J server instance to be imported.

- Select the **Import** box next to each shared library to import. To use the latest installed version of the library, do not specify a version number.
- Specify a minimum or maximum version of a shared library to import.

Use this feature to import a different version of a shared library than that inherited from the application's parent. For example, you could import an earlier version of the Oracle JDBC driver than that inherited from the OC4J `default` application by specifying the maximum version to import.

- De-select the **Import** box for any shared library that should not be inherited from the parent application.

To remove all inherited shared libraries, deselect the **Inherit parent application's imported shared libraries** checkbox. This action adds the following `<remove-inherited>` element to the `orion-application.xml` file that will be generated for the application at deployment time:

```
<imported-shared-libraries>
  <remove-inherited name="*" />
</imported-shared-libraries>
```

- Optionally specify additional code sources to add as library paths to the OC4J instance.

To add a code source, specify either a relative or absolute path or URL to the archive. Specified directories are scanned for archives to load at OC4J server startup.

- Manage the classloader created for each Web module.

OC4J creates a classloader instance for each Web module deployed as a WAR into the server instance.

Check the **Search Local Classes First** checkbox to force the class loader to also load JARs containing classes and resources packaged within the WAR before loading JARs external to the WAR.

For example, suppose you want to ensure that your Web module uses the version of `log4j` packaged within your WAR, and not use the version of `log4j` bundled with a resource adapter deployed into the OC4J instance. Selecting this option forces the classloader to load the local `log4j` JAR file contained within the WAR.

If the WAR contains a `MANIFEST.MF`, you can force JARs declared as named extensions to be loaded by checking the **Include WAR Manifest Class Path** box. You can also specify the relative or absolute paths to one or more additional code sources containing classes or resources to be loaded in the **Classpath** field. Using either of these features causes classes to be loaded by the Web module's classloader, just as if they were packaged within the WAR.

## Configuring Clustering

The OC4J clustering framework supports replication of objects and values contained in an HTTP session or a stateful session Enterprise JavaBean instance across OC4J instances.

By default, applications inherit the clustering configuration set at the parent application level. However, clustering can also be configured at the application level at deployment time. A configuration property set at the application level overrides the corresponding value inherited from the parent application.

- Select **Enable Clustering** to enable clustering support for the application. The value selected overrides the setting inherited from the application's parent.

If clustering has been enabled at the parent application level, the parent's configuration will be applied by default.

- Select **Peer-to-Peer Replication** to enable the peer-to-peer replication mechanism. Note that the mechanism used is dependent on the type of OC4J installation.

- OC4J instances within Oracle Application Server

In an Oracle Application Server cluster, dynamic peer-to-peer is used to enable OC4J instances to dynamically discover and communicate with one another.

Dynamic peer-to-peer is used by default if clustering is enabled; no additional configuration is needed.

Note that you can optionally specify the IP address for a 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.

- Standalone OC4J installation

In a standalone configuration, each node in the cluster can be statically configured to recognize at least one other peer node. As a node becomes aware of each of its peers, it also becomes aware of each peer's peer(s)—with the end result that all of the nodes in the cluster become aware of one another.

- Select **Multicast Replication** to configure OC4J to send and receive HTTP session and stateful session bean state changes via multicast packages.

This is the default replication protocol used in a standalone OC4J installation. The multicast address and port must be the same for all instances in the cluster. The OC4J default address is `230.230.0.1`; the default port is `45566`.

Note that you can optionally specify the IP address for a 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

- Select **Database Replication** to replicate application state to a database.

Select the JNDI name of the data source providing the connection to the database from the **Database JNDI Location** pull-down menu. See the *Oracle Containers for J2EE Services Guide* for details on creating and using data sources.

## Providing Resource Mappings

Any resources included in the EAR file being deployed will be displayed in the Application Server Control Console. Map any environment references in your application to physical entities currently present in the operational environment to JMS topics and queues, data sources or resource adapters.

Map any references to resources in your application, such as data sources or mail queues, to physical entities currently present in the OC4J container. Note that if you need a specific resource, you must have already added this to the OC4J container before you deploy your application in order for you to match them in this step.

For most applications, the resource reference you must designate is the data source JNDI name. You cannot configure the data source information through Deployment Tasks in Application Server Control; it only designates an already configured data source or a data source that you will be configuring later. Designate the JNDI location name of the data source that the application will be using.

If you have any MDBs in your EAR file, you may be required to add information about the subscriptions or topics. If you are defining `DataSource` objects for CMP entity beans, you are given the option to add a JNDI location for those `DataSource` objects.

## Undeploying an Application

Undeploying an application or module removes the code from the OC4J runtime and deletes all existing Web site bindings.

1. Click the **Applications** tab ->**Undeploy** button.
2. Select the application, then click the **Undeploy** button.

---

---

## Deploying with the OC4J Ant Tasks

OC4J provides a set of Ant tasks for performing deployment-related operations on a specific OC4J instance or on a "group" of instances within a cluster. This chapter describes the Ant tasks and provides guidelines for integrating the tasks into your application build process.

This chapter includes the following sections:

- [Preparing to Use the Ant Tasks](#)
- [Invoking the OC4J Ant Tasks](#)

---

---

**Note:** The OC4J Ant tasks discussed in this chapter are intended to be used with Apache Ant version 1.6.2.

See the following link to access the most recent Apache Ant product documentation:

<http://ant.apache.org/manual/>

---

---

### Preparing to Use the Ant Tasks

This section provides prerequisites and guidelines for using the OC4J Ant tasks. It includes the following topics:

- [Overview of the Ant Tasks](#)
- [Prerequisites for Using the Ant Tasks](#)
- [Incorporating the OC4J Ant Tasks into Your Environment](#)
- [Enabling Logging](#)
- [Sample build.xml File](#)

### Overview of the Ant Tasks

Ant 1.6.2 is installed with OC4J in the `ORACLE_HOME/ant/` directory structure. The `oracle-ant.jar` is installed by default within the `ORACLE_HOME/ant/lib` directory.

The following Ant-related files are installed with OC4J in the `ORACLE_HOME/j2ee/utilities` directory:

- `ant-oracle-classes.jar`  
A JAR file containing the compiled Ant task classes.
- `ant-oracle.properties`

An editable properties file used to specify the execution properties for the Ant tasks. You may want to edit this file to reflect your environment.

- `ant-oracle.xml`

An XML file that can be optionally imported into the Ant build file (`build.xml`) using the Ant `<import>` task. This is only necessary if `oracle-ant.jar` is not installed in the `ORACLE_HOME/ant/lib` directory.

## Prerequisites for Using the Ant Tasks

The following prerequisites are required to use the deployment-related OC4J Ant tasks as outlined in this documentation.

- Ant version 1.6.2 or later must be used.
- An `ORACLE_HOME` environment variable set to the OC4J installed directory must be defined.
- A `JAVA_HOME` environment variable set to the location of the Java2 Standard Edition SDK must be defined.

## Incorporating the OC4J Ant Tasks into Your Environment

This section outlines the procedure for setting up your build environment to use the Ant 1.6.2 implementation installed with OC4J by default in `ORACLE_HOME/ant`.

1. Add `ORACLE_HOME/ant/bin` to the system PATH.
2. Declare the `oracle` namespace in the `<project>` element in the Ant build file (`build.xml`). The OC4J Ant tasks will be referenced in `build.xml` using this namespace.

```
<project name="test" default="all" basedir="."
  xmlns:oracle="antlib:oracle">
```

3. (OPTIONAL) Copy the `ant-oracle.properties` file from the `ORACLE_HOME/j2ee/utilities` directory to the directory containing your build file (`build.xml`).

Although you can modify the file in `ORACLE_HOME/j2ee/utilities` and reference it from your build scripts, it is better to maintain the original file as a template.

4. (OPTIONAL) Set the values for arguments to pass to the Ant tasks in the `ant-oracle.properties` file.

The properties within the file are set to the OC4J default values. The file also reads in environment variable settings, such as `ORACLE_HOME` and `JAVA_HOME`. You can edit any of these properties as necessary to reflect the configuration of the target OC4J instance.

5. (OPTIONAL) If you copied the `ant-oracle.properties` file to your build directory, you must reference it in the build script (`build.xml`). For example:

```
<property file="ant-oracle.properties"/>
```

## Incorporating the Ant Tasks Using Ant 1.6.2 Outside OC4J

This section outlines the procedure for setting up your build environment to use the Ant 1.6.2 implementation outside OC4J.

1. Add `ORACLE_HOME/ant/bin` to the system PATH.



2. Set the `ANT_HOME` environment variable to point to your Ant installation.
3. Declare the `oracle` namespace in the `<project>` element in the Ant build file (`build.xml`). The OC4J Ant tasks will be referenced in `build.xml` using this namespace.

```
<project name="test" default="all" basedir="."
  xmlns:oracle="antlib:oracle">
```

4. Copy the `ant-oracle.properties` file from the `ORACLE_HOME/j2ee/utilities` directory to the directory containing your build file (`build.xml`).

Although you can modify the file in `ORACLE_HOME/j2ee/utilities` and reference it from your build scripts, it is better to maintain the original file as a template.

5. Set the values for arguments to pass to the Ant tasks in the `ant-oracle.properties` file.

The properties within the file are set to the OC4J default values. The file also reads in environment variable settings, such as `ORACLE_HOME` and `JAVA_HOME`. You can edit any of these properties as necessary to reflect the configuration of the target OC4J instance.

6. Copy the `ant-oracle.xml` from the `ORACLE_HOME/j2ee/utilities` directory to the directory containing your build file (`build.xml`).
7. Import `ant-oracle.xml` into your build script (`build.xml`). For example:

```
<property file="ant-oracle.xml"/>
```

## Enabling Logging

You can enable Java logging to help troubleshoot errors that occur when running the Ant tasks. Log messages will be output to the console.

To enable logging:

1. Create an `ANT_OPS` environment variable and set the value to `-Djava.util.logging.config.file=logging.properties` before running the Ant tasks.
2. 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/ant/bin`, you must include the path to the file in the `ANT_OPS` variable.

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

**Table 10-1 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 10–1 (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
```

## Sample build.xml File

The following is an example `build.xml` file containing a single deploy task. This task will deploy the specified EAR to a standalone OC4J server.

```
<project name="test" default="deploy" basedir="." xmlns:oracle="antlib:oracle">
  <property name="lib.dir" value="/scratch/temp"/>
  <property name="app.name" value="hello-planet"/>
  <property name="deployer.uri" value="deployer:oc4j:localhost:23791"/>
  <property name="oc4j.admin.user" value="oc4jadmin"/>
  <property name="oc4j.admin.password" value="password"/>
  ...
  <target name="deploy-ear" depends="setup,check-oc4j-available>
    <echo message="----> Deploying the application module deployment (ear) file"/>
    <oracle:deploy deployerUri="${deployer.uri}"
      userId="${oc4j.admin.user}"
      password="${oc4j.admin.password}"
      file="${lib.dir}/${app.name}.ear"
      deploymentName="${app.name}"
      bindAllWebApps="default-web-site"
      logFile="${log.dir}/deploy-ear.log"/>
  </target>
  ...
</project>
```

## Invoking the OC4J Ant Tasks

This section provides guidelines on using the deployment-related Ant tasks provided with OC4J.

Each task is specified within a `<target>` element in the build file (`build.xml`), in a subelement formatted as `<oracle:taskName. . . />` Note that `oracle` is the namespace used to reference the OC4J Ant tasks. See ["Sample build.xml File"](#) on page 10-4 for a sample build file.

See the following sections for guidelines on invoking each task:

- [Setting the Deployer URI](#)
- [Deploying an EAR File](#)
- [Deploying a Standalone WAR File](#)

- [Deploying a Standalone RAR File](#)
- [Binding All Web Modules to a Single Web Site](#)
- [Binding a Specific Web Module to a Specific Web Site and Setting the Context URI](#)
- [Undeploying an Archive](#)
- [Updating Modified Classes Only in a Deployed EJB Module](#)
- [Installing a Shared Library](#)
- [Starting or Stopping an Application](#)

## Setting the Deployer URI

The key property passed to an Ant task `deployerUri`, which specifies the target OC4J instance(s) for the task. The syntax for the URI varies depending on the instance(s) being targeted.

See the following for the format of this URI:

- [Invoking a Task on a Group of OC4J Instances Within a Cluster](#)
- [Invoking a Task on a Specific OC4J Instance Within a Cluster](#)
- [Invoking a Task on a Standalone OC4J Server](#)

### Invoking a Task 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 synchronized 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/home
```

**Table 10–2** URI Parameters for Targeting a Group

Parameter	Description
<code>rmis</code>	Optional. Include if the target utilizes ORMI over SSL, or ORMIS.
<code>host</code>	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.
<code>opmnPort</code>	Optional. The OPMN request port, as specified in <code>opmn.xml</code> . If not specified, the default port 6003 will be used.
<code>oc4jInstanceName</code>	Required. The common instance name shared by OC4J instances within the group.

### Invoking a Task on a Specific OC4J Instance Within a Cluster

Use the following URI to target a specific OC4J 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:6015/instance2/home
```

**Table 10–3 URI Parameters for Targeting a Specific Instance**

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

### Invoking a Task 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 10–4 URI Parameters for Targeting Standalone OC4J**

Parameter	Description
<code>rmis</code>	Required if the target utilizes ORMI over SSL, or ORMIS.
<code>host</code>	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.
<code>rmiPort</code>	Required if RMI used. The RMI port, as specified in the instance-specific <code>rmi.xml</code> file.

**Table 10–4 (Cont.) URI Parameters for Targeting Standalone OC4J**

Parameter	Description
<i>ormisPort</i>	Required if ORMS used. The SSL port, as specified in the instance-specific <i>rmi.xml</i> file.

## Deploying an EAR File

Use the `deploy` task to deploy a J2EE application packaged in an EAR file. The following example shows the properties typically supplied to deploy an EAR file.

```
<oracle:deploy
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  file="${lib.dir}/${app.name}.archiveType"
  deploymentName="${app.name}"
  bindAllWebApps="default-web-site"
  deploymentPlan="localPath/filename"
  logFile="${log.dir}/deploy-ear.log"/>
```

The following table summarizes the properties that can be passed to the task.

**Table 10–5 deploy Properties for EAR Deployment**

Parameter	Description
<code>deployerUri</code>	The URI specifying the deployment target.
<code>userId</code>	The administrator user name for the target OC4J instance.
<code>password</code>	The administrator password for the target OC4J instance.
<code>file</code>	The path and file name of the archive to deploy.
<code>deploymentName</code>	The user-defined application deployment name, used to identify the application within OC4J.
<code>bindAllWebApps</code>	Binds all Web modules to the specified Web site. Specify the <i>name</i> portion of the <i>name_web-site.xml</i> file that configures the Web site.
<code>deploymentPlan</code>	Optional. The path and file name 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.
<code>parent</code>	The parent application of this application. The default is the global or default application.
<code>targetPath</code>	Optional. The directory to deploy the EAR to. If not specified, the EAR is deployed to the <code>ORACLE_HOME/j2ee/home/applications/</code> directory by default. The deployed EAR file is also copied to this directory. Each successive deployment will cause this EAR file to be overwritten.
<code>deploymentDirectory</code>	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 <code>ORACLE_HOME/j2ee/home/application-deployments/</code> .

**Table 10–5 (Cont.) deploy Properties for EAR Deployment**

Parameter	Description
enableIIOP	<p>Optional. Include to generate IIOP client stubs on the OC4J server.</p> <p>The application-level stubs generated for all EJB modules are output to an archive named <code>_iiopClient.jar</code> in the <code>/j2ee/home/application-deployments/appName</code> directory. In addition, stubs for each individual EJB module are generated in an archive with the same name in the <code>/j2ee/home/application-deployments/appName/ejbModule/</code> directory.</p> <p>Note that the <code>GenerateIIOP</code> system property must be enabled at OC4J startup to use this feature. This property is set as <code>-DGenerateIIOP=true</code> on the OC4J command line for OC4J standalone or as an <code>oc4j-options</code> value in <code>opmn.xml</code>.</p>
iiopClientJarPath	<p>Optional. The path and filename of the JAR to output IIOP client stubs to.</p> <p>The application-level stubs generated for all EJB modules are output to an archive named <code>_iiopClient.jar</code> in the <code>/j2ee/home/application-deployments/appName</code> directory. If a path is supplied, the archive is also set on this path.</p> <p>In addition, stubs for each individual EJB module are generated in an archive with the same name in the <code>/j2ee/home/application-deployments/appName/ejbModule/</code> directory.</p> <p>Note that the <code>GenerateIIOP</code> system property must be enabled at OC4J startup to use this feature. This property is set as <code>-DGenerateIIOP=true</code> on the OC4J command line for OC4J standalone or as an <code>oc4j-options</code> value in <code>opmn.xml</code>.</p>
logFile	The path and name for a log file generated for the deployment.

## Deploying a Standalone WAR File

Use the `deploy` task to deploy a standalone Web module packaged in a WAR file.

```
<oracle:deploy
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  file="${lib.dir}/${app.name}.war"
  deploymentName="${app.name}"
  bindAllWebApps="default-web-site"
  logFile="${log.dir}/deploy-war.log"/>
```

The following table summarizes the WAR-specific properties that can be passed to the task.

**Table 10–6 deploy Properties for Standalone WAR Deployment**

Property	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.
password	The administrator password for the target OC4J instance.
file	The path and file name of the archive to deploy.

**Table 10–6 (Cont.) deploy Properties for Standalone WAR Deployment**

Property	Description
deploymentName	The user-defined module deployment name, used to identify the module within OC4J.
bindAllWebApps	Binds the Web module to the specified Web site. Specify the <i>name</i> portion of the <i>name_web-site.xml</i> file that configures the Web site.
deploymentPlan	Optional. The path and file name 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.
parent	Optional. The parent application of this module. The default is the global or default application.
targetPath	Optional. The directory to deploy the archive to. If not specified, the archive is deployed to the <i>ORACLE_HOME/j2ee/home/applications/</i> directory by default. The deployed archive file is also copied to this directory. Each successive deployment will cause this file to be overwritten.
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 <i>ORACLE_HOME/j2ee/home/application-deployments/</i> .
contextRoot	The Web module context, which will be appended to the URL used to access the application through a Web browser. For example, if you supply <i>/petstore</i> as the context root, the module could be accessed with the following URL: <code>http://node1.company.com:7777/petstore</code>
logFile	The path and name for a log file generated for the deployment.

## Deploying a Standalone RAR File

The `deploy` task deploys a J2EE application or a standalone module packaged in an archive. The following example shows the properties typically supplied to deploy a standalone RAR file.

```
<oracle:deploy
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  file="${lib.dir}/${app.name}.rar"
  deploymentName="${app.name}"
  grantAllPermissions="true"
  logFile="${log.dir}/deploy-rar.log"/>
```

The following table summarizes the properties that can be passed to the task.

**Table 10–7 deploy Task Properties for Standalone RAR Deployment**

Parameter	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.
password	The administrator password for the target OC4J instance.

**Table 10–7 (Cont.) deploy Task Properties for Standalone RAR Deployment**

Parameter	Description
file	The path and file name of the archive to deploy.
deploymentName	The user-defined connector name, used to identify the connector within OC4J.
deploymentPlan	Optional. The path and file name 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.
nativeLibPath	Optional. The path to the directory containing native libraries (such as DLLs) within the RAR file.
grantAllPermissions	Include and set to true to grant all runtime permissions requested by the resource adapter, if required.
logFile	The path and name for a log file generated for the deployment.

## Redeploying an Archive

The `redeploy` task can be used to redeploy a previously-deployed archive. Note that the previous version of the archive will be undeployed as part of this process.

The syntax is as follows:

```
<oracle:redeploy
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  file="${lib.dir}/${app.name}.archiveType"
  deploymentName="${app.name}"
  keepsettings="true"
  sequential="true"
  logFile="${log.dir}/deploy-ear.log"/>
```

**Table 10–8 redeploy Task Properties**

Subswitch	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.
password	The administrator password for the target OC4J instance.
file	Required. The path and file name of the archive to redeploy.
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.



**Table 10–8 (Cont.) redeploy Task Properties**

Subswitch	Description
<code>sequential</code>	<p>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.</p> <p>If not included, the archive is simultaneously deployed to all instances by default.</p> <p>This option is valid in a clustered environment only; it is not valid for standalone OC4J.</p>

## Binding All Web Modules to a Single Web Site

The `bindAllWebApps` task binds the Web modules within a previously deployed EAR to a specified Web site

```
<oracle:bindAllWebApps
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  deploymentName="${app.name}"
  webSiteName="${oc4j.binding.module}"
```

The following table summarizes the properties that can be passed to the task.

**Table 10–9 bindAllWebApps Task Properties**

Property	Description
<code>deployerUri</code>	The URI specifying the deployment target.
<code>userId</code>	The administrator user name for the target OC4J instance.
<code>password</code>	The administrator password for the target OC4J instance.
<code>deploymentName</code>	The user-defined name of the application that the Web modules belong to, set when the application was deployed.
<code>webSiteName</code>	The name of the <code>name_web-site.xml</code> file that denotes the Web site that this Web application should be bound to.

## Binding a Specific Web Module to a Specific Web Site and Setting the Context URI

The `bindWebApp` task binds a specific Web module to the Web site that will be used to access it. You can also specify the context URI for the module.

```
<oracle:bindWebApp
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  deploymentName="${app.name}"
  webModule="${web.name}"
  webSiteName="${oc4j.binding.module}"
  contextRoot="/${context.root}"/>
```

The following table summarizes the properties that can be passed to the task.

**Table 10–10** *bindWebApp Task Properties*

Property	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.
password	The administrator password for the target OC4J instance.
deploymentName	The user-defined name of the application the Web module belongs to, set when the application was deployed.
webModule	The name of the Web module to be bound to the Web site. This should be the name of the WAR file contained within the EAR file, without the .WAR extension.
webSiteName	The name of the <i>name_web-site.xml</i> file that denotes the Web site that this Web application should be bound to.
contextRoot	The context URI for the Web module, such as /utility. This will be appended to the URL used to access the application through a Web browser; for example <code>http://localhost:8888/utility</code> .

## Undeploying an Archive

Use the `undeploy` task to remove an application or module from the OC4J instance(s). Note that the `isConnector="true"` property must be included if you are undeploying a standalone resource adapter (RAR).

```
<oracle:undeploy
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  deploymentName="${app.name}"
  logFile="${log.dir}/filename.log"/>
```

The following table summarizes the properties that can be passed to the task.

**Table 10–11** *undeploy Task Properties*

Property	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.
password	The administrator password for the target OC4J instance.
deploymentName	The user-defined name of the application or module to undeploy. This is the name set when the archive was deployed.
isConnector	Include and set to <code>true</code> if undeploying a standalone RAR.
logFile	The path and name for a log file generated for the deployment.

## Updating Modified Classes Only in a Deployed EJB Module

The `updateEJBModule` task 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.

```
<oracle:updateEJBModule
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
```

```
password="${oc4j.admin.password}"
deploymentName="${app.name}"
ejbModuleName="${ejb.jar}"
file="${new.ejb.jar}"
logFile="${log.dir}/filename.log"/>
```

The following table summarizes the properties that can be passed to the task.

**Table 10–12** *updateEJBModuleTask Properties*

Property	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.
password	The administrator password for the target OC4J instance.
deploymentName	Required. The name of the application the EJB is part of. If you are updating a standalone EJB module, specify the default application.
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.
logFile	The path and name for a log file generated for the update.

## Installing a Shared Library

Use the `publishSharedLibrary` task to install a shared library within a single OC4J instance. Once installed, the shared library will be available for use by applications within the instance.

---

**Note:** In the current release, this task can only be run against a single standalone or OPMN-managed OC4J instance. It will not create a shared library across multiple instances with a cluster.

---

The shared library binaries will be installed in the `ORACLE_HOME/j2ee/instance/shared-lib` directory within the OC4J instance. At the same time, a `<shared-library>` element declaring the shared library will be added to the `server.xml` file on the OC4J instance.

```
<oracle:publishSharedLibrary
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  libraryName="name"
  libraryVersion="version"
  logFile="${log.dir}/filename.log">
  <oracle:uploadCodeSource path="path/file" />
  <oracle:addCodeSource path="path/file" />
  <oracle:sharedLibraryImport libraryname="name" min-version="version"
    max-version="version" />
</oracle:publishSharedLibrary>
```

- To upload a new code source to the OC4J server, specify the path and file name of the JAR or ZIP archive file to upload in a nested `<oracle:uploadCodeSource>` element. The path can be absolute or relative to the current working directory.

- To add a JAR or ZIP that already exists on the server, specify the path and file name in an `<oracle:addCodeSource>` element. Specify an absolute or relative path pointing to the location of the existing file on the OC4J server. Note if a relative path is used, it will be interpreted as relative to `ORACLE_HOME`.
- To import an existing shared library into the new shared library, specify the shared library name as defined within the OC4J instance in an `<oracle:sharedLibraryImport>` element. Note that you can optionally specify the minimum and/or maximum version of the library to import.

Include one element for each code source to upload or add. Do the same for each existing shared library to import.

The following example uploads two JAR files to the target OC4J server:

```
<oracle:publishSharedLibrary
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  libraryName="acme.common"
  libraryVersion="2.5"
  logFile="${log.dir}/filename.log">
  <oracle:uploadCodeSource path="/acme/acme-apis.jar" />
  <oracle:uploadCodeSource path="/acme/acmeImpl.jar" />
</oracle:installSharedLibrary>
```

The following table summarizes the properties that can be passed to the task.

**Table 10–13** *publishSharedLibraryTask Properties*

Property	Description
<code>deployerUri</code>	The URI specifying the deployment target.
<code>userId</code>	The administrator user name for the target OC4J instance.
<code>password</code>	The administrator password for the target OC4J instance.
<code>libraryName</code>	Required. The name of the shared library. In cases where common APIs are implemented by multiple vendors, the name should include both the vendor name and the name of the technology; for example, <code>oracle.jdbc</code> or <code>xerces.xml</code> .
<code>libraryVersion</code>	Required. The shared library version. This value should ideally reflect the code implementation version.
<code>parentName</code>	Optional. The name of the parent shared library, if applicable.
<code>parentVersion</code>	Optional. The parent shared library version, if applicable.
<code>logFile</code>	The path and name for a log file generated for the update.

## Modifying an Existing Shared Library

Use the `modifySharedLibrary` task to make changes to an existing shared library installed within a single OC4J instance.

---



---

**Note:** In the current release, this task can only be run against a single standalone or OPMN-managed OC4J instance. It will not create a shared library across multiple instances with a cluster.

---



---

```
<oracle:modifySharedLibrary
```

```

deployerUri="${deployer.uri}"
userId="${oc4j.admin.user}"
password="${oc4j.admin.password}"
libraryName="name"
libraryVersion="version"
logFile="${log.dir}/filename.log">
  <oracle:uploadCodeSource path="path/file" />
  <oracle:removeCodeSource path="path/file" />
  <oracle:addCodeSource path="path/file" />
  <oracle:addImport libraryName="name" min-version="version
    max-version="version" />
  <oracle:removeImport libraryname="name" min-version="version
    max-version="version" />
</oracle:modifySharedLibrary>

```

- To upload a new code source to the OC4J server, specify the path and file name of the JAR or ZIP archive file to upload in a nested `<oracle:uploadCodeSource>` element. The path can be absolute or relative to the current working directory.
- To add a JAR or ZIP that already exists on the server, specify the path and file name in an `<oracle:addCodeSource>` element. Specify an absolute or relative path pointing to the location of the existing file on the OC4J server. Note if a relative path is used, it will be interpreted as relative to `ORACLE_HOME`.
- Use `<oracle:removeCodeSource>` to remove an existing code source from the shared library. Specify the location of the code source in the `ORACLE_HOME/j2ee/instance/shared-lib` directory within the OC4J instance.
- To import an existing shared library into the shared library, specify the shared library name as defined within the OC4J instance in an `<oracle:addImport>` element. Note that you can optionally specify the minimum and/or maximum version of the library to import.
- To remove an imported shared library, use an `<oracle:removeImport>` element.

Include one element for each code source to upload, add or remove. Do the same for each existing shared library to import or remove.

The following example removes a code source and an imported library from the target shared library:

```

<oracle:modifySharedLibrary
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  libraryName="acme.common"
  libraryVersion="2.5"
  logFile="${log.dir}/filename.log">
  <oracle:removeCodeSource path="c:/oracle/j2ee/home/shared-lib/acme-apis.jar" />
  <oracle:removeImport libraryName="foo" min-version="2.0"/>
</oracle:installSharedLibrary>

```

The following table summarizes the properties that can be passed to the task.

**Table 10–14** *modifySharedLibraryTask Properties*

Property	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.

**Table 10–14 (Cont.) modifySharedLibraryTask Properties**

Property	Description
password	The administrator password for the target OC4J instance.
libraryName	Required. The name of the shared library to affect. In cases where common APIs are implemented by multiple vendors, the name should include both the vendor name and the name of the technology; for example, <code>oracle.jdbc</code> or <code>xerces.xml</code> .
libraryVersion	Required. The shared library version. This value should ideally reflect the code implementation version.
logFile	The path and name for a log file generated for the update.

## Starting or Stopping an Application

You can use the start or stop task to start, stop or restart an application and its child applications as part of a deployment operation on a specific OC4J instance or across an entire cluster.

```
<oracle:start | stop
  deployerUri="${deployer.uri}"
  userId="${oc4j.admin.user}"
  password="${oc4j.admin.password}"
  deploymentName="${app.name}"
```

The following table summarizes the properties that can be passed to the task.

**Table 10–15 start/stopTask Properties**

Property	Description
deployerUri	The URI specifying the deployment target.
userId	The administrator user name for the target OC4J instance.
password	The administrator password for the target OC4J instance.
deploymentName	Required. The name of the application to start or stop.

---

---

## Deploying with the `admin_client.jar` Utility

OC4J provides a command-line utility— `admin_client.jar`—that can be used to perform deployment-related 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` 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.

- [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 table below.

```
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 command(s) supplied. The syntax for the URI varies depending on the instance(s) being targeted. See the following for the format of this URI:

- [Deploying to a Group of OC4J Instances Within a Cluster](#)
- [Deploying to a Specific OC4J Instance](#)
- [Deploying to 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
```

### Deploying to 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 synchronized 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, in which cluster nodes are aware of one another. You only need to supply the host name and optionally the 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://opmnHost[:opmnPort]/oc4jInstanceName
```

For example:

```
deployer:cluster:opmn://node1/home
```

**Table 11–1** URI Parameters for Targeting a Group

Parameter	Description
<code>rmis</code>	Optional. Include if the target utilizes ORMI over SSL, or ORMIS.
<code>opmnHost</code>	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.
<code>opmnPort</code>	Optional. The OPMN request port, as specified in <code>opmn.xml</code> . If not specified, the default port 6003 will be used.



**Table 11–1 (Cont.) URI Parameters for Targeting a Group**

Parameter	Description
<i>oc4jInstanceName</i>	Required. The common instance name shared by OC4J instances within the group.

### Deploying to 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 11–2 URI Parameters for Targeting a Specific Instance**

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

### Deploying to 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 11–3 URI Parameters for Targeting Standalone OC4J**

Parameter	Description
<i>rmis</i>	Required if the target utilizes ORMI over SSL, or ORMIS.

**Table 11–3 (Cont.) URI Parameters for Targeting Standalone OC4J**

Parameter	Description
<i>host</i>	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.
<i>rmiPort</i>	Required if RMI used. The RMI port, as specified in the instance-specific <code>rmi.xml</code> file.
<i>ormisPort</i>	Required if ORMIS used. The SSL port, as specified in the instance-specific <code>rmi.xml</code> file.

### Validating a URI

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

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

For example:

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

### Printing Usage Text to the Console

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

```
java -jar admin_client.jar -help
```

### 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/j2ee/home/`, 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 11–4 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 11–4 (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]
```

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 11–5 -deploy Command Subswitches for EAR Deployment**

Subswitch	Description
-file	Required. The path and file name 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 page 11-9.  You can optionally supply a value for <i>webSiteName</i> , which is the <i>name</i> portion of the <i>name_web-site.xml</i> file that configures the Web site.
-targetPath	Optional. The directory to deploy the EAR to. If not specified, the EAR is deployed to the <i>ORACLE_HOME/j2ee/instance/applications/</i> 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 <i>ORACLE_HOME/j2ee/instance/application-deployments/</i> .
-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 <i>_iiopClient.jar</i> in the <i>ORACLE_HOME/j2ee/home/application-deployments/ appName /directory</i> . In addition, stubs for each individual EJB module are generated in an archive with the same name in the <i>ORACLE_HOME/j2ee/home/application-deployments/ appName/ ejbModuleName/</i> directory.  Note that the <i>GenerateIIOP</i> system property must be enabled at OC4J startup to use this feature. This property is set as <i>-DGenerateIIOP=true</i> on the OC4J command line for OC4J standalone or as an <i>oc4j-options</i> value in <i>opmn.xml</i> .

**Table 11–5 (Cont.) -deploy Command Subswitches for EAR Deployment**

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

## 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 must 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 `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 11–6 -deploy Command Subswitches for WAR Deployment**

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

**Table 11–6 (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 <i>webSiteName</i> , which is the <i>name</i> portion of the <i>name_web-site.xml</i> file that configures the Web site.
-targetPath	Optional. The directory to deploy the archive to. If not specified, the archive is deployed to the <i>ORACLE_HOME/j2ee/instance/applications/</i> 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 <i>ORACLE_HOME/j2ee/instance/application-deployments/</i> .
-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 <i>-deploymentName</i> will be used.  For example, if you supply <i>/petstore</i> as the context root, the module could be accessed with the following URL:  <code>http://node1.company.com:7777/petstore</code>

## 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 11-14 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 11–7 -deploy Command Subswitches for RAR Deployment**

Subswitch	Description
-file	Required. The path and file name 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	Optional. 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.

This section covers the following topics:

- [Binding All Web Modules to a Single Web Site](#)
- [Binding a Specific Web Module to a Specific Web Site and Setting the Context Root](#)

### Binding 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 the `default-web-site` by default.

The syntax is:

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

**Table 11–8 -bindAllWebApps Command Subswitches**

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

### Binding a Specific Web Module to a Specific Web Site and Setting 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 -bindAllWebApps
-appName appName -webModuleName moduleName -webSiteName siteName
-contextRoot contextRoot
```

**Table 11–9 -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 <i>name_web-site.xml</i> file that denotes the Web site to bind the Web module to. If omitted, all modules are bound to <i>default-web-site</i> .
-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 <code>http://localhost:8888/petstore</code> .  If not supplied, the context root specified in the parent application's <i>application.xml</i> 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 11–10 -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 11-14 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 `-updateEJBModule` command 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 11–11** *-updateEJBModule Syntax*

Option	Description
<code>-appName</code>	Required. The name of the application the EJB is part of. If updating a standalone EJB module, specify the <code>default</code> application.
<code>-ejbModuleName</code>	Required. The name of the EJB JAR file to be updated as defined in <code>application.xml</code> .

**Table 11–11 (Cont.) -updateEJBModule Syntax**

Option	Description
-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 11–12** *-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, <code>oracle.jdbc</code> or <code>xerces.xml</code> .
-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 11–13** *-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 11–13 (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.

The syntax is as follows:

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

The following example starts the `petstore` application on `node2` within the cluster:

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

---

---

## Deploying to Standalone OC4J with admin.jar

---

---

**Note:** The `admin_client.jar` utility is the recommended option for all deployment and management operations, and should be used in place of `admin.jar`.

---

---

The following chapter provides instruction on using the `admin.jar` command-line utility provided with OC4J, which can be used to deploy or undeploy J2EE applications to a standalone OC4J instance. The following topics are covered:

- [Understanding the admin.jar Syntax](#)
- [Deploying/Redeploying an Application](#)
- [Undeploying an Application](#)
- [Updating an EJB Module Within a Deployed Application](#)
- [Deploying/Redeploying a Standalone Connector](#)
- [Undeploying a Standalone Connector](#)

Note that only those `admin.jar` options for deployment and undeployment are documented in this chapter. See the *Oracle Containers for J2EE Configuration and Administration Guide* for complete instructions on using the `admin.jar` utility.

---

---

**Notes:**

- `admin.jar` cannot be used to deploy 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`.
- 
- 

### Understanding the admin.jar Syntax

The `admin.jar` utility uses the following syntax. The variables are described in [Table 12-1](#).

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

For example, the following command will print the `admin.jar` help to the console. 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 -help
```

**Table 12–1 Setting the Host and Login Information**

Variable	Description
<code>oc4jHost:oc4jOrmiPort</code>	<p>The host name and port of the OC4J server on which you are invoking <code>admin.jar</code>.</p> <p>The <code>admin.jar</code> 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 <code>rmi.xml</code> file for the OC4J server to which you are directing the request.</p> <p>The 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 <code>rmi.xml</code> file in the <code>&lt;rmi-server&gt;</code> element, as follows:</p> <pre>&lt;rmi-server port="oc4jOrmiPort" host="oc4jHost" /&gt;</pre>
<code>adminId</code> <code>adminPassword</code>	<p>The OC4J administration user name and password. The user name for the default administrator account is <code>oc4jadmin</code>.</p>

## Deploying/Redeploying an Application

The `-deploy` option is used to deploy or redeploy a J2EE application packaged in an EAR file into an OC4J instance. Note that OC4J must already be running before `admin.jar` can be used.

1. Open a command console and change to the `J2EE_HOME` directory.
2. Deploy the archive into OC4J. The syntax is as follows:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
adminPassword -deploy -file path/filename
-deploymentName appName [-targetPath path] [-parent appName]
[-deploymentDirectory path] [-iiopClientJar path/filename]
```

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
```

The following table provides details on the `-deploy` subswitches:

**Table 12–2 -deploy Command Subswitches**

Subswitch	Description
<code>-file</code>	Required. The path and file name of the EAR file to deploy.
<code>-deploymentName</code>	Required. The user-defined application deployment name, used to identify the application within OC4J.

**Table 12–2 (Cont.) -deploy Command Subswitches**

Subswitch	Description
-targetPath	Optional. The directory to deploy the EAR to. If not specified, the EAR is deployed to the <code>ORACLE_HOME/j2ee/instance/applications/</code> 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.
-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 <code>ORACLE_HOME/j2ee/instance/application-deployments/</code> .
-iiopClientJar	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 file name of the JAR to output the generated stubs to. Otherwise, copies of the stubs will be output to an archive named <code>_iiopClient.jar</code> in a new subdirectory with the same name as the deployed EJB JAR in <code>ORACLE_HOME/j2ee/instance/application-deployments/</code> .  Note that the <code>GenerateIIOP</code> system property must be enabled at OC4J startup to use this feature. For example:  <code>java -DGenerateIIOP=true -jar oc4j.jar</code>

3. Next, bind the application to the Web site that will be used to access it. The syntax is:

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

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 admin password -bindwebapp utility
    utility-web default-web-site /utility
```

**Table 12–3 -bindWebApp Command Parameters**

Parameter	Description
appName	The user-defined name of the application, which is the same name used for <code>-deploymentName</code> in the <code>-deploy</code> option.
webAppName	The name of the Web module. This should be the name of the WAR file contained within the EAR file, without the <code>.WAR</code> extension.
webSiteName	The name of the <code>&lt;name&gt;_web-site.xml</code> file that denotes the Web site that this Web application should be bound to.
contextRoot	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, <code>http://localhost:8888/utility</code> .

## Undeploying an Application

The following command removes an application from the OC4J runtime and removes bindings from any Web sites to which the application's Web modules were bound.

1. Open a command console and change to the `ORACLE_HOME` directory.
2. Undeploy the application. The syntax is as follows:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminpassword
    -undeploy appName
```

- `appName` is the application name, which must match the value specified for `-deploymentName` on the `-deploy` option.

---

**Note:** The optional `-keepFiles` subswitch, which could be used to prevent files from being removed from the installed directories, has been deprecated. All files are now removed during undeployment.

---

For example, the following undeploys the utility application:

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

## Updating an EJB Module Within a Deployed Application

The `admin.jar` command-line utility provided with OC4J includes an `-updateEJBModule` option that allows an updated EJB JAR to be redeployed to an application running within an OC4J instance. Only those beans that have changed within the EJB JAR will be deployed.

This option is intended to be used by an application developer to redeploy the JAR file directly from his/her development environment. For more information on using `admin.jar`, see the *Oracle Containers for J2EE Configuration and Administration Guide*.

The syntax for `-updateEJBModule` is as follows:

```
java -jar admin.jar ormi://oc4j_host:oc4j_ormi_port admin_id
    admin_password -application appName -updateEJBModule ejbJarName
    [-file path/ejbJarName]
```

Usage notes:

- Specify the application the EJB is a component of as the value for `appName`. This name must match the name specified at deployment.
- If the updated EJB JAR file is in the working directory, and its location matches the relative module path defined in the application's `application.xml` J2EE deployment descriptor, you only need to specify the EJB JAR file name as the value for `ejbJarName`.
- If the updated EJB JAR is *not* in the working directory, or is in a subdirectory that does not match the relative module path defined in `application.xml`, specify the JAR file's location using the optional `[-file]` subswitch.

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 at the root level of the `/dist` directory and the relative path defined in `application.xml` is “`customerEjb.jar`”, the following command could be issued from the `/dist` directory:

```
java -jar $J2EE_HOME/admin.jar ormi://myoc4jserver:23791 oc4jadmin password
      -application petstore -updateEJBModule customerEjb.jar
```

However, if the updated file is located within the `/build` directory, the optional `-file` subswitch can be used to specify this location:

```
java -jar $J2EE_HOME/admin.jar ormi://myoc4jserver:23791 oc4jadmin password
      -application petstore -updateEJBModule customerEjb.jar
      -file build/customerEjb.jar
```

---



---

#### Notes:

- The examples assume that a `J2EE_HOME` environment variable pointing to the `oc4j_installed_dir/j2ee/home` directory within the target OC4J host exists on the developer’s machine.
  - An error will occur if the EJB module name is missing or invalid, or if the updated EJB JAR cannot be found.
- 
- 

## Deploying/Redeploying a Standalone Connector

Use the `-deployconnector` command to deploy or redeploy a standalone Java Connector Architecture-compliant resource adapter packaged in a RAR file. Note that redeploying a standalone resource adapter requires a restart of the `default` application.

During redeployment of a resource adapter packaged within a RAR file, all existing application components will continue to obtain connection factories from the existing version of the resource adapter. New components, however, will obtain connection factories from the newly deployed resource adapter.

Existing JCA connections will remain open until closed by the application; new connections will be created from the original resource adapter instance.

The syntax is as follows:

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
      adminPassword -deployconnector -file path -name connectorName
      [-nativeLibPath path] [-grantAllPermissions]
```

**Table 12–4** *-deployconnector* Command Subswitches

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

## Undeploying a Standalone Connector

Use `-undeployconnector` to remove a standalone connector from the OC4J runtime. The default application must be restarted after a standalone resource adapter is undeployed.

The syntax is as follows. Note that the connector name must be supplied.

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId  
adminPassword -undeployconnector -name connectorName
```

---

---

## Deploying Web Applications from Eclipse

This chapter explains how to deploy a Web application to a standalone OC4J server directly from Eclipse using the Web Tools Platform, a set of Eclipse plug-ins provided under the aegis of the Eclipse Foundation.

The Eclipse Web Tools Platform project extends the Eclipse platform with tools for developing J2EE Web applications.

### Installing the Web Tools Platform

Note that these instructions assume that you have already installed and configured a standalone OC4J instance. It also assumes that you have already installed Eclipse 3.1 for Windows.

Download and install each of the following files into the `<eclipse_home>/eclipse/plugin` directory. You can download all of these required components from the following Eclipse site:

<http://download.eclipse.org/webtools/downloads/>

- Eclipse Modeling Framework (EMF), `emf-sdo-xsd-SDK-2.1.0.zip`
- Graphical Editing Framework (GEF), `GEF-SDK-3.1.zip`
- Java EMF Model Runtime (JEM), `JEM-SDK-1.1.zip`
- Java EMF Model Runtime (JEM), `JEM-SDK-1.1.zip`
- Eclipse Web Tools Platform Project (WTP), `wtp-1.0M8.zip`

### Deploying a Web Application with the Web Tools Platform

This section provides an overview of how to create and deploy a Web application to OC4J using the Eclipse Web Tools Platform.

#### Connecting to OC4J from Eclipse

This section shows you how to connect to a standalone OC4J instance from Eclipse.

1. Launch Eclipse.
2. Open the J2EE perspective and select **Window > Open Perspective > Other** menu. Choose **J2EE** in the resulting dialog. This perspective includes the Servers view, which is used in the example deployment.
3. Select **Window > Show View > Console** menu. This enables you to see the server output.

4. Right-click in the Servers View and select **New > Server** menu. This launches the New Server wizard.
5. In the Define a New Server panel, **Select Oracle > Generic Oracle OC4J Standalone Server 10.1.3**. Click **Next**.
6. Set the following in the Define a New Generic Oracle OC4J Standalone 10.1.3 Runtime panel:
  - For JRE, select the JDK you are using.
  - For Oracle J2EE Home, browse to the `\j2ee\home` subdirectory in the folder where you installed the Oracle Application Server.
  - Click **Next**.
7. Set the password to the password for the `oc4jadmin` administrator account you created during the OC4J installation in the Create a new Generic Oracle OC4J Standalone 10.1.3 server panel. Accept the defaults for all the other fields.
8. Click **Finish**.

## Building a Web Application

Next, create a simple Web application to deploy to the OC4J server instance.

1. From the ProjectExplorer view, right-click on the **Dynamic Web Projects** folder and select **New Dynamic Web Project**.
2. Enter a name for the project.
3. Click the Show Advanced button. Note that the Target runtime field is pre-populated with the entry for OC4J: Generic Oracle OC4J Standalone 10.1.3.
4. Accept the default values for all the fields in this dialog and click **Finish**.
5. Now you will create a JSP page within your new project. Open the **Dynamic Web Projects** folder.
6. Expand the project you created and then the **WebContent** folder.
7. Right-click on the **WebContent** folder and select the **New JSP** menu.
8. Enter a name for the file, such as `index.jsp`, and click **Finish**.
9. The file is opened in a JSP Editor. Enter this text between the `<BODY>` tags:

```
<% out.print("Hello World!!"); %>
```
10. Save the file.

## Deploying a Web Application

Once the Web application is ready, it can be deployed to OC4J directly from Eclipse.

1. Right-click on the `jsp` file, `index.jsp`, in the ProjectExplorer. Select **Run As > Run on Server** menu.
2. In the Run On Server dialog, verify that the server Oracle OC4J Standalone Server v10.1.3 is selected.
3. Click **Finish**. The Eclipse WTP tool will now do the following:
  - Package the Web application.
  - Start the OC4J server if it is not running.

- Publish the application to the OC4J instance.
- Launch the application in a browser.

The console view displays the log tracking the progress of the deployment.



---

---

## Automatic Deployment in OC4J

This chapter discusses automatic deployment functionality in OC4J, which enables you to automatically reload only modified files within an application to an OC4J instance, rather than requiring that the entire application be redeployed.

This chapter includes the following topics:

- [What Is Automatic Deployment in OC4J?](#)
- [Using an "Auto-Deployment" Directory](#)
- [Using the Check For Updates Feature](#)
- [Forcing a One-Time Redeployment Using admin.jar](#)

### What Is Automatic Deployment in OC4J?

Automatic deployment, or *OC4J polling*, is a task management feature that automatically checks for changes made to currently deployed applications and modules, and reloads those files that have been modified. This functionality is a tremendous benefit for developers, eliminating the need to go through the deployment process every time code is updated.

By default, OC4J checks for files to deploy every second. This interval is configurable through the `taskmanager-granularity` attribute of the `<application-server>` element in the `server.xml` configuration file. See the *Oracle Containers for J2EE Configuration and Administration Guide* for details on task manager configuration.

In addition to automatic polling, the `admin.jar` command-line utility includes an `-updateConfig` option that forces OC4J to check for updated files. You can use this feature in a production environment to check for and reload updated files on an as-needed basis. See "[Forcing a One-Time Redeployment Using admin.jar](#)" on page 14-5 for details on this feature.

### When Should Automatic Deployment Be Used?

Automatic deployment is recommended only for standalone OC4J instances in a development environment. It is not recommended for use in production environments.

The reason is that the polling mechanism is invoked by the task manager on a regular schedule and uses system resources. In addition, automatic deployment carries the risk of putting OC4J in an inconsistent state, and errors may result if requests try to execute against OC4J.

## Using an "Auto-Deployment" Directory

Automatic deployment can be initiated by dropping an EAR file in a designated "auto-deployment" directory within the OC4J instance. Note this feature should only be used in a standalone OC4J development environment.

The directory must be created on the server hosting the OC4J instance; it is not created by OC4J. An existing directory within OC4J, such as *ORACLE\_HOME/j2ee/home/applications*, can also be used.

The location of the directory must then be specified in the `application-auto-deploy-directory` attribute, which must be added to the root `<application-server>` element in *ORACLE\_HOME/j2ee/home/config/server.xml*.

The `server.xml` entry below sets *ORACLE\_HOME/j2ee/home/applications* as the auto-deployment directory:

```
<application-server ...
  application-directory=" ../applications"
  check-for-updates="adminClientOnly"
  deployment-directory=" ../application-deployments"
  application-auto-deploy-directory=" ../applications">
  ...
</application-server>
```

---

---

**Note:** If the check for updates feature is not enabled, OC4J must be restarted for configuration changes made in `server.xml` to take effect.

---

---

Once configured, OC4J will poll the directory for new or updated EAR files every time the task manager is executed. The server compares the timestamp on an EAR file to determine if a redeployment should be initiated. If it should, the EAR will be deployed automatically. Any Web modules packaged as WAR files within the EAR will be bound automatically to the `default` Web site.

The auto-deployment directory feature is completely independent of OC4J polling. Archives dropped in this directory will be deployed regardless of whether OC4J polling is enabled or disabled.

## Using the Check For Updates Feature

The "check for updates" feature allows you to redeploy files to an OC4J instance, as the following topics describe:

- [Enabling/Disabling Check For Updates](#)
- [What Types of Files Are Automatically Redeployed?](#)
- [What Happens During Redeployment?](#)

---

---

**Note:** An EAR or WAR file copied to the *ORACLE\_HOME/j2ee/home/applications* directory will be deployed or redeployed by default upon OC4J startup, regardless of whether auto-deployment is enabled.

---

---



## Enabling/Disabling Check For Updates

The check for updates feature can be enabled or disabled through the following methods:

- The `check-for-updates` attribute of the root `<application-server>` element in `ORACLE_HOME/j2ee/home/config/server.xml`. For example:

```
<application-server ... check-for-updates="all" ... />
```

- Setting the `checkForUpdates` system property on the `oc4j.jar` command line. For example:

```
java -DcheckForUpdates=all -jar oc4j.jar
```

---

**Notes:** The following notes apply to the `checkForUpdates` system property:

- All system properties are prefaced on the command line with a `-D`.
  - The value set for this property overrides the value set in `server.xml`.
- 

Table 14–1 contains the values that can be set for `checkForUpdates` using either option.

**Table 14–1 Valid Values for `checkForUpdates`**

Value	Description
all	<p>Enables OC4J polling, which starts automatically at the interval specified in the OC4J task manager configuration. The default interval is every 1 minute.</p> <p>This option should not be used in Oracle Application Server or production environments.</p> <p>This value also allows the <code>-updateConfig</code> option to be passed on the <code>admin.jar</code> command line, which forces OC4J to perform a one-time check for updated files.</p> <p>See <a href="#">"Forcing a One-Time Redeployment Using admin.jar"</a> on page 14-5 for details on this feature.</p>
adminClientOnly	<p>This is the default value set in both standalone OC4J and Oracle Application Server installations. It allows the <code>-updateConfig</code> option to be passed on the <code>admin.jar</code> command line, which forces OC4J to do a one-time check for updated files, and reload any that have changed.</p> <p>See <a href="#">"Forcing a One-Time Redeployment Using admin.jar"</a> on page 14-5 for details on this feature.</p>
none	<p>Completely disables OC4J polling, including the <code>-updateConfig</code> option.</p>

## What Types of Files Are Automatically Redeployed?

The following files can be automatically redeployed to an OC4J instance:

- Modified OC4J-specific XML configuration files in the `ORACLE_HOME/j2ee/home/config/` directory, including `server.xml`.
- Modified deployment descriptors packaged in an updated EAR file copied to the `ORACLE_HOME/j2ee/home/applications/` directory.

- The following files packaged within an updated WAR file. The WAR can be packaged in an EAR file copied to the `ORACLE_HOME/j2ee/home/applications/` directory, or copied directly to the Web module's `ORACLE_HOME/j2ee/home/applications/<webAppName>/` directory.
  - Modified deployment descriptors
  - Updated files in the `WEB-INF/lib/` or `WEB-INF/classes/*` paths within the WAR
  - Updated JSP tag library (TLD) files

Note that this feature does not currently provide automatic detection of EJB or data source-related configuration changes. This means, for example, that modified files in an EJB JAR file will not be automatically redeployed. OC4J must be restarted to detect such configuration changes and apply them appropriately.

## What Happens During Redeployment?

The following tables describe the impact of modifying or updating various files when `checkForUpdates` is set to `all`, indicating that the feature is enabled. See ["Enabling/Disabling Check For Updates"](#) on page 14-3 for instructions on enabling OC4J polling.

[Table 14–2, "Impact of Modifying OC4J Configuration Files"](#) describes the impact of modifying OC4J configuration files within the `ORACLE_HOME/j2ee/home/config/` directory in an OC4J instance.

**Table 14–2 Impact of Modifying OC4J Configuration Files**

Modified File	Action Initiated
<code>server.xml</code>	Modifying the OC4J server configuration file causes the OC4J server to be restarted.
<code>global-web-application.xml</code>	Modifying this file, used to configure OC4J servlet and JSP containers, forces all Web modules bound to all Web sites within the instance to be restarted.
<code>application.xml</code>	Modifying this file, which contains common settings for all applications in this OC4J instance, forces all deployed applications to be restarted.
<code>*-web-site.xml</code>	Modifying a Web site configuration file causes the Web site to be restarted. Incoming requests will not be serviced during the restart.

[Table 14–3, "Impact of Modifying Files in an EAR"](#) describes the impact of modifying one or more deployment descriptors within an updated EAR file copied to the `ORACLE_HOME/applications/` directory.

**Table 14–3 Impact of Modifying Files in an EAR**

Modified File	Action Initiated
<code>application.xml</code>	Modifying the J2EE standard application deployment descriptor within an EAR forces a restart of the application.

**Table 14–3 (Cont.) Impact of Modifying Files in an EAR**

Modified File	Action Initiated
OC4J deployment descriptors	Modifying an OC4J proprietary deployment descriptor packaged within a deployed EAR, such as <code>orion-application.xml</code> , causes the EAR to be redeployed and then restarted with the updated configuration.

Table 14–4, "Impact of Modifying Files in a WAR" describes the impact of modifying files or deployment descriptors within an updated WAR file deployed as part of an application. The updated WAR file can either be packaged in an EAR or be copied to the Web module's `ORACLE_HOME/j2ee/home/applications/<webAppName>/` directory.

OC4J checks the timestamp of each file and redeploys only those that have a different timestamp from the other files.

**Table 14–4 Impact of Modifying Files in a WAR**

Modified File	Action Initiated
<code>web.xml</code> or <code>orion-web.xml</code>	Modifying either the J2EE standard Web deployment descriptor or the OC4J specific descriptor within an updated WAR file causes the Web module to be restarted with the new configuration.
<code>WEB-INF/lib/</code>	Updating a library JAR file in this path forces the Web module to be restarted and modified classes to be reloaded.
<code>WEB-INF/classes/*</code>	Updating a file in this path forces the Web module to be restarted and modified classes to be reloaded.
<code>.tld</code> files	Updating a TLD file forces the Web module to be restarted.

## Forcing a One-Time Redeployment Using admin.jar

The `admin.jar` command-line utility can be used to administer a standalone OC4J instance. This tool includes an `-updateConfig` option that enables OC4J polling on an as-needed basis, forcing OC4J to check its directories for updated files and reload any that have changed.

---



---

### Usage Notes:

- The `admin.jar` tool can only be used against a standalone OC4J instance. It cannot be used against an OPMN-managed instance.
  - `checkForUpdates` must be set to either `all` or `adminClientOnly` to use this feature.
- 
- 

The utility is installed in `ORACLE_HOME/j2ee/home` by default. Note that the OC4J server must be started before this utility can be used.

The syntax for this command is as follows:

```
java -jar admin.jar ormi://<host>:<ormiPort> <adminId> <adminPassword>
-updateConfig
```

In the example below, 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.

```
cd ORACLE_HOME\j2ee\home
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -updateConfig
```

See the *Oracle Containers for J2EE Configuration and Administration Guide* for additional instructions on using admin.jar.

---

---

## Troubleshooting Deployment Errors

This chapter discusses common errors that may occur during deployment. It includes the following sections:

- [Interruptions During Application Deployment](#)
- [Exceptions During Application Deployment](#)

### Interruptions During Application Deployment

If the deployment process is interrupted for any reason, you may need to clean up the temp directory, which by default is `/var/tmp`, on your system.

The Application Server Control Console deployment wizard uses approximately 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 of its additional files.

However, if the wizard is interrupted, it may not have the time or opportunity to clean up the temp directory. Thus, you must clean up any additional deployment files from this directory yourself. Otherwise, this directory may fill up, which will disable any further deployment.

You can change the temp directory at OC4J startup by setting the `java.io.tmpdir` command-line option to a new location. See the *Oracle Containers for J2EE Configuration and Administration Guide* for details on setting system properties.

### Exceptions During Application Deployment

This section provides details on the following types of errors that may occur during deployment:

- [OC4J Out of Memory Errors](#)
- [Java Compiler Out of Memory Errors](#)
- [Stack Overflow Errors](#)
- ["Too many open files" Errors](#)

#### OC4J Out of Memory Errors

Deploying a large EAR file - such as a file larger than 75 MB - may cause OC4J to throw `java.lang.OutOfMemory` errors. If sufficient memory is available, you can eliminate this problem by increasing the heap size for the OC4J process at OC4J startup. For example:

```
java -Xms512m -Xmx512m -jar oc4j.jar
```

This problem may also be encountered when deploying an application using the `admin.jar` command-line utility. Again, the solution is to increase the heap size for this utility:

```
java -Xms512m -Xmx512m -jar admin.jar ormi://localhost:23791 admin welcome -deploy
...
```

If running under Unix/Linux, verify that the `ulimit` settings allow the JVM process to allocate this much memory.

## Java Compiler Out of Memory Errors

OC4J may return the following message when using the `javac` compiler in out-of-process mode when compiling EJB wrapper classes:

```
The system is out of resources.
Consult the following stack trace for details.
java.lang.OutOfMemoryError: Java heap space
```

This message indicates that the external JVM process spawned to execute the `javac` compiler has run out of memory.

The default heap size allocated to the compiler is 1024 MB. To allocate more memory to the compiler, increase the heap size by setting the `-Xmx` option through the `options` attribute of the `<java-compiler>` element in `server.xml`. For example:

```
<java-compiler name="javac" in-process="false" options="-J-Xmx2048m"/>
```

## Stack Overflow Errors

The `javac` compiler may throw a `java.lang.StackOverflowError` when compiling EJB wrapper classes if your application is too large.

The thread stack size option enables the control of the `stacksize` attribute of a thread attributes object. This attribute specifies the minimum stack size to be used for the created thread.

If this occurs, you can increase the thread stack size by setting the `-Xss` option through the `options` attribute of the `<java-compiler>` element in `server.xml`. For example:

```
<java-compiler name="javac" in-process="false" options="-J-Xmx2048m -J-Xss=8m"/>
```

## "Too many open files" Errors

When deploying large applications, the OC4J JVM may throw "too many open files" exceptions. For example:

```
java.net.SocketException: Too many open files
java.io.IOException: Too many open files
```

These exceptions indicate that the operating system has run short of file descriptors, which are used by processes to identify open files of different types, including sockets or pipes.

Increasing the number of file descriptors will typically resolve this kind of problem. On Unix platforms, this can be performed using the `ulimit -n` command. The maximum number of file descriptors, as well as the maximum size that can be allocated to a process, are defined by a resource limit. Refer to your operating

system-specific system administration manuals for instructions on setting the hard limit values.





---

---

## 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 commercial 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

Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

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

Licensors 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 MERCHANTABILITY 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](mailto: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](mailto: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 MERCHANTABILITY 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](mailto: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](mailto: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.



## A

---

admin\_client.jar tool  
  using, 11-1  
admin.jar  
  deploying with, 12-1  
admin.jar tool  
  deploying with, 12-2  
  undeploying with, 11-11, 12-4  
Ant  
  deploying with, 10-1  
application  
  deployment overview, 1-1  
  undeploying, 2-5  
application mount points  
  configuring, 2-2  
Application Server Control Console  
  deploying to clusters with, 9-2  
  deploying with, 9-1  
applications  
  automatic redeployment, 14-1  
  deploying, 11-5  
  undeploying, 11-5  
auto-deployment, 14-1  
auto-deployment directory, 14-2  
automatic redeployment, 14-1

## B

---

binding an application, 2-2

## C

---

clustering configuration, 9-7  
clusters  
  deploying to, 9-2  
configuration  
  setting JSP configuration parameters, 5-3

## D

---

deployable components, 1-1  
deployment  
  error recovery, 15-1  
deployment descriptors  
  and deployment plans, 8-2  
  overview, 8-2

deployment plans  
  and packaged deployment descriptors, 8-2  
  creating, 8-3  
  list of properties, 8-4  
  overview, 8-1  
deployment process  
  overview, 2-1  
deployment tasks, 9-4  
  configuring clustering, 9-7  
  configuring EJBs, 9-5  
  managing shared libraries, 9-6  
  mapping roles, 9-5  
  resource mappings, 9-8  
  selecting the security provider, 9-4  
deployment tools  
  admin.jar, 12-1  
  Ant, 10-1  
  Application Server Control Console, 9-1  
  JDeveloper, 1-4

## E

---

Eclipse  
  deploying Web applications with, 13-1  
EJB module  
  deploying, 4-1  
  impact of redeployment on clients, 4-4  
  incremental deployment, 4-1  
  updating, 11-11  
entity beans  
  redeploying, 4-4

## G

---

groups  
  deploying to, 9-2, 10-5

## I

---

incremental EJB deployment, 4-1

## J

---

JDeveloper  
  deploying with, 1-4  
JSPs

adding to a deployed application, 5-2  
modifying in a deployed application, 5-2

## **M**

---

mod\_oc4j, 2-2  
mount points  
    configuring for deployed applications, 2-2

## **O**

---

OC4J  
    administration, 11-1  
    command-line interface, 11-1  
OC4J polling, 14-1  
    disabling, 14-3  
Oracle HTTP Server, 2-2

## **P**

---

parent application  
    designating, 2-1  
polling, 14-1

## **R**

---

resource adapters  
    deploying, 12-5  
    undeploying, 12-5

## **S**

---

scripted deployment, 10-1, 12-1  
    using admin.jar, 1-3  
    using Ant tasks, 1-3  
session beans  
    redeploying, 4-4

## **U**

---

undeployment, 2-5

## **V**

---

valid archive types, 1-1

## **W**

---

WAR files  
    deploying as standalone modules, 5-1  
    wrapped within an EAR file, 5-1  
Web module  
    adding JSPs to a, 5-2  
    deploying, 5-1  
    modifying JSPs in a, 5-2  
    redeploying, 5-1  
Web services  
    deploying, 7-1  
    redeploying, 7-1  
Web site  
    binding an application to a, 2-2

Web Tools Platform, 13-1

## **X**

---

XML schemas  
    viewing, 8-2  
XSDs  
    viewing, 8-2