Oracle® Application Server

Release Notes

10*g* Release 3 (10.1.3) for AIX 5L Based Systems (64-Bit)

B28074-06

April 2009

ORACLE®

Oracle Application Server Release Notes, 10*g* Release 3 (10.1.3) for AIX 5L Based Systems (64-Bit)

B28074-06

Contents

3 General Management and Security Issues

4 Oracle HTTP Server

5   Oracle Containers for J2EE

6   Oracle Application Server Portal

7   Oracle TopLink

8   Oracle Business Rules

9   OracleAS Disaster Recovery

10   Oracle Sensor Edge Server

Preface

This preface includes the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This document is intended for users of Oracle Application Server 10*g*.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at

`http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Related Documents

For more information, see these Oracle resources:

- Oracle Application Server Documentation on Oracle Application Server Disk 1
- Oracle Application Server Documentation Library 10*g* Release 3 (10.1.3)

Conventions

The following text conventions are used in this document:

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

What's New in the *Oracle Application Server Release Notes*?

This chapter provides a listing of new topics introduced with this version of the *Oracle Application Server Release Notes*. The new topics are in the following chapters:

- Chapter 2, "Installation and Upgrade Issues"
- Chapter 3, "General Management and Security Issues"
- Chapter 4, "Oracle HTTP Server"
- Chapter 5, "Oracle Containers for J2EE"

Chapter 2, "Installation and Upgrade Issues"

- Section 2.3.4, "PATH, CLASSPATH, and LD_LIBRARY_PATH May be Set or Unset When Running Installer"

Chapter 3, "General Management and Security Issues"

- Section 3.3.10, "Incorrect request Attribute Description"
- Section 3.3.11, "Incorrect Title"

Chapter 4, "Oracle HTTP Server"

- Section 4.1.3, "Changing the Location of the PID File Requires a Change in apachectl"
- Section 4.1.4, "Routing Requests to Different Middle Tiers Based on the URL of the Request"
- Section 4.2.9, "Clarification for the Name of the Oracle Application Server Proxy Plug-In Definition File"

Chapter 5, "Oracle Containers for J2EE"

- Section 5.2.10, "Using a Tag Library with the ojspc Utility"
- Section 5.9.6.4, "Configuration of OPMN Gateway Element Shows Incorrect Order"
- Section 5.9.6.5, "Starting Order Incorrect for Static Peer-to-Peer Replication"

1

Introduction

This chapter introduces Oracle Application Server Release Notes, 10*g* Release 3 (10.1.3). It includes the following topics:

- Section 1.1, "Latest Release Information"
- Section 1.2, "Purpose of this Document"
- Section 1.3, "Operating System Requirements"
- Section 1.4, "Certification Information"
- Section 1.5, "Licensing Information"

1.1 Latest Release Information

This document is accurate at the time of publication. Oracle will update the release notes periodically after the software release. You can access the latest information and additions to these release notes on the Oracle Technology Network at:

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

1.2 Purpose of this Document

This document contains the release information for Oracle Application Server 10*g* Release 3 (10.1.3). It describes differences between Oracle Application Server 10*g* Release 3 (10.1.3) and its documented functionality.

Oracle recommends you review its contents before installing, or working with the product.

1.3 Operating System Requirements

Oracle Application Server installation and configuration will not complete successfully unless users meet the hardware and software pre-requisite requirements before installation. Refer to *Oracle Application Server Installation Guide* for a complete list of operating system requirements.

1.4 Certification Information

The latest certification information for Oracle Application Server 10*g* Release 3 (10.1.3) is available at:

https://metalink.oracle.com

1.5 Licensing Information

Licensing information for Oracle Application Server 10*g* Release 3 (10.1.3) is available at:

http://oraclestore.oracle.com

Detailed information regarding license compliance for Oracle Application Server 10*g* Release 3 (10.1.3) is available at:

http://www.oracle.com/technology/products/ias/index.html

2

Installation and Upgrade Issues

This chapter describes installation and upgrade issues and their workarounds associated with Oracle Application Server. It includes the following topics:

- Section 2.1, "Installation Issues"
- Section 2.2, "Upgrade Issues"
- Section 2.3, "Documentation Errata"

2.1 Installation Issues

This section describes issues with installation of Oracle Application Server. It includes the following topics:

- Section 2.1.1, "sl_DlgClusterInfoWebReturn Parameter Does Not Work in Silent Installations"
- Section 2.1.2, "Response Files Created through the Record Mode Are Not Supported"
- Section 2.1.3, "IPv6 Not Supported"

2.1.1 sl_DlgClusterInfoWebReturn Parameter Does Not Work in Silent Installations

Oracle Application Server Installation Guide 10g Release 3 (10.1.3) lists `sl_DlgClusterInfoWebReturn` as one of the parameters that can be included in the response file for silent installations. This parameter is supposed to allow you add the instance to a cluster; however, it does not work and you will find that the instance is not clustered after the installation has completed.

To work around this problem, you can add the instance to a cluster by manually updating the `opmn.xml` file with the following tags after you finish the silent installation:

```
<topology>
    <discover list="*ip_address:port"/>
</topology>
```

These tags must be nested within the `<notification-server>` tag.

For more detailed instructions, refer to *Oracle Application Server High Availability Guide 10g Release 3 (10.1.3)*, section 3.2.1.2, "Discovery Server Method".

2.1.2 Response Files Created through the Record Mode Are Not Supported

If you want to run the installer with a response file (to perform a silent or non-interactive installation), you cannot use response files created through the -record command-line option in the installer, as described in the "Creating Response Files by Using the Record Mode in the Installer" section in the *Oracle Application Server Installation Guide*. Instead, you must use response files that were created from the provided response file templates; you replace the placeholder values in the template files with your own values.

2.1.3 IPv6 Not Supported

This release of Oracle Application Server is not certified to run on machines that are configured with IPv6. You have to install and run this release of Oracle Application Server on machines that are configured with IPv4.

2.2 Upgrade Issues

This section describes issues with upgrade of Oracle Application Server. It includes the following topic:

- Section 2.2.1, "Additional Data Source Requirement for OEMS JMS Database Applications"

2.2.1 Additional Data Source Requirement for OEMS JMS Database Applications

If you are deploying an OEMS JMS Database application on Oracle Application Server 10g Release (10.1.3), note that you must verify that the manage-local-transactions attribute in the data-sources.xml file is set to false.

The following example shows the managed-data-source element in the data-sources.xml file with the required attribute for OEMS JMS Database applications:

```
<managed-data-source name="OracleDS" connection-pool-name="Example
Connection Pool" jndi-name="jdbc/OracleDS" *manage-local-transactions="false"*/>.
```

2.3 Documentation Errata

This section describes known errors in Oracle Application Server installation and upgrade documentation. It includes the following topic:

- Section 2.3.1, "Incorrect Reference to Installation Media"
- Section 2.3.2, "Incorrect Ports Listed in the Oracle Application Server Installation Guide"
- Section 2.3.3, "ORACLE_HOSTNAME Environment Variable Does Not Work"
- Section 2.3.4, "PATH, CLASSPATH, and LD_LIBRARY_PATH May be Set or Unset When Running Installer"

2.3.1 Incorrect Reference to Installation Media

In the Oracle Application Server Quick Installation Guide, the second step in section 5 "Starting the Installer" should be:

2. Insert the disk:

CD-ROM: Insert the Oracle Application Server CD-ROM.

DVD-ROM: Insert the Oracle Application Server DVD-ROM.

2.3.2 Incorrect Ports Listed in the *Oracle Application Server Installation Guide*

In Table B-1, "Default Port Numbers and Ranges (Grouped By Component)", in the *Oracle Application Server Installation Guide*, the allotted port range is incorrectly listed for the Listen (SSL) and SSL ports. The correct port numbers are as follows:

For the Listen (SSL) port:

■ Allotted port range: 4443 - 4543

■ Default port number: 4443

For the SSL port:

■ Allotted port range: 4443 - 4543

■ Default port number: 4443

2.3.3 ORACLE_HOSTNAME Environment Variable Does Not Work

In the section "Installing on Multihomed (Multi-IP) Computers," the *Oracle Application Server Installation Guide* states:

"By default, Oracle Universal Installer uses the ORACLE_HOSTNAME environment variable setting to find the hostname. If ORACLE_HOSTNAME is not set and you are installing on a computer that has multiple network cards, Oracle Universal Installer determines the hostname by using the first name in the /etc/hosts file."

This is incorrect. There is no default behavior, and the only way to set the hostname is by running Oracle Universal Installer with the OUI_HOSTNAME switch as follows:

```
> runInstaller OUI_HOSTNAME=hostname
```

2.3.4 PATH, CLASSPATH, and LD_LIBRARY_PATH May be Set or Unset When Running Installer

The "Environment Variables" section of *Oracle Application Server Installation Guide 10g Release 3 (10.1.3)* states that the PATH, CLASSPATH, and LD_LIBRARY_PATH environment variables must not contain references to directories in any Oracle home directories.

This is not correct. It does not matter, if the PATH, LD_LIBRARY_PATH, and CLASSPATH variables are set or unset before running the installer, and it does not matter if they contain references to directories in any Oracle home directories.

3

General Management and Security Issues

This chapter describes management and security issues associated with Oracle Application Server. It includes the following topics:

- Section 3.1, "General Issues and Workarounds"
- Section 3.2, "Clustering and Replication Issues"
- Section 3.3, "Documentation Errata"

3.1 General Issues and Workarounds

This section describes general management and security issues. It includes the following topics:

- Section 3.1.1, "Deploying an Application Hangs During File Upload"
- Section 3.1.2, "Limited Management Support for Multiple-JVM OC4J Instances"
- Section 3.1.3, "OC4J Restart Required When Changing the Name or URL of a JDBC Data Source or Connection Pool"
- Section 3.1.4, "Problem Removing a Property from a Native Data Source"
- Section 3.1.5, "Important Restriction When Setting Thread Pool Size on the Thread Pool Configuration Page"
- Section 3.1.6, "Use the Cluster Topology Page to Restart the OC4J Instance"
- Section 3.1.7, "TopLink Sessions Not Available in Application Server Control Console"
- Section 3.1.8, "Unable to Receive MBean Notification Using OPMN to Start or Stop OC4J"
- Section 3.1.9, "Using the Java Server Pages Standard Tag Libraries"
- Section 3.1.10, "Error While Generating Web Service"
- Section 3.1.11, "Problem with Deployment of non-English Character Java Server Pages"
- Section 3.1.12, "Additional Step Required When Configuring ODL Logging for Oracle HTTP Server"

3.1.1 Deploying an Application Hangs During File Upload

When you deploy an application using Application Server Control, the upload page may keep showing the "upload in progress" message until you manually refresh the page. In the `ASControl.log` file, you may see these messages:

```
WARN  util.MultipartRequestUtil _getMultiPartRequestParameter.263 -
 java.io.IOException: Cannot create temp directory: The system cannot find the
 path specified
```

You get this message if the temporary directory specified by the `java.io.tmpdir` system property does not exist.

To determine the value of the `java.io.tmpdir` system property, go to the Application Server Control's System Property page, which lists the values of system properties. If the directory specified by the `java.io.tmpdir` system property does not exist, create the directory and restart the server.

3.1.2 Limited Management Support for Multiple-JVM OC4J Instances

With Oracle Application Server 10*g* Release 3 (10.1.3), you can configure OC4J to use multiple Java Virtual Machines (JVMs) by setting the `numprocs` argument in the `opmn.xml` file to a number greater than one (1).

For example:

```
<ias-component id="OC4J">
   <process-type id="home" module-id="OC4J" status="enabled">
         .
         .
         .
      <process-set id="default_group" numprocs="2"/>
   </process-type>
</ias-component>
```

However, this feature is not supported by Oracle Enterprise Manager 10*g* Application Server Control (Application Server Control). Instead, when the `numprocs` argument is set to more than one (1), you must use command line tools to manage your Oracle Application Server environment. For example, you must use:

- `admin_client.jar` for deployment, re-deployment, undeployment, start and stop applications, and shared library management

- Apache Ant for deployment, redeployment, and undeployment of your applications

- `opmnctl` commands for starting, stopping, and other life cycle operations on the Oracle Application Server

For all other administrative configuration changes, if you are using multiple JVMs, you must shut down Oracle Application Server, manually configure the relevant XML files, and then restart Oracle Application Server.

3.1.3 OC4J Restart Required When Changing the Name or URL of a JDBC Data Source or Connection Pool

If you modify the name or the connection URL of a JDBC data source or JDBC connection pool, then you must restart the OC4J instance; otherwise the changes you make will not take effect.

For example, if you use the JDBC Resources page in the Application Server Control Console to change the connection URL of a JDBC connection pool, you will not be prompted to restart the OC4J instance, but the restart is required. If you do not restart the OC4J instance, any deployed applications that require the data source will attempt to use the original connection URL.

> **See Also:** "Managing Data Sources and JDBC Connection Pools" in the Application Server Control online help

3.1.4 Problem Removing a Property from a Native Data Source

If you use the Application Server Control Console to remove a property from a native data source, Enterprise Manager does not remove the property from the underlying connection factory. As a result, the property (and its current value) is not changed.

This is expected behavior. To set a value on the underlying connection factory, use the `setProperty` operation of the `JDBCDataSource` MBean for the native Data Source to do this. You can use the MBean Browser, which is available in the Application Server Control Console, to invoke an MBean operation.

> **See Also:** "About the MBean Browser" in the Application Server Control online help

3.1.5 Important Restriction When Setting Thread Pool Size on the Thread Pool Configuration Page

By default, the Thread Pool Configuration page in the Application Server Control Console shows a minimum thread pool size of one (1) for an unconfigured thread pool. This value is expected, but do not click **OK** on this screen with a minimum thread pool size set to one (1) for any thread pool.

If you are configuring OC4J thread pools using the Thread Pool Configuration Administration Task, be sure to replace the default value for the minimum thread pool size to a number that is greater than 10; otherwise, you will not be able to login to the Application Server Control Console the next time you restart the OC4J instance.

In the event that you set the minimum thread pool size to 1 and then click **OK** on the Thread Pool Configuration page, and as a result, you cannot access the Application Server Control Console, do the following:

1. Stop the OC4J instance using the command line.

   For example, on a UNIX system, in a managed, Oracle Application Server environment, enter the following command:

   `ORACLE_HOME/opmn/bin/opmnctl stopall`

   > **See Also:** "Starting and Stopping" in the *Oracle Application Server Administrator's Guide*
   >
   > "Starting and Stopping OC4J" in the *Oracle Containers for J2EE Configuration and Administration Guide*

2. Edit the `server.xml` file so that all thread pools are configured with a minimum pool size that is greater than 10.

   The server.xml for the OC4J instance is located in the following directory:

   `ORACLE_HOME/OC4J_instance_name/config/`

   For example:

   `ORACLE_HOME/home/config/`

Alternatively, to reset thread pool configuration to the default to factory settings, remove the `<global-thread-pool>` element and the `<work-manager-thread-pool>` element from the `server.xml` file.

3.1.6 Use the Cluster Topology Page to Restart the OC4J Instance

Some OC4J configuration pages in the Application Server Control Console (including the JTA Administration and Oracle Internet Directory Association pages) require a restart of the OC4J instance for changes to take affect.

If you use the **Restart** link, which is displayed after applying changes to one of these pages, the operation may take a few minutes because it performs an internal restart of the OC4J instance. Instead of using the **Restart** link, Oracle recommends that users navigate to the Cluster Topology page, select the affected OC4J instance, and then click **Restart** to perform a full restart of the OC4J instance. In a standalone OC4J environment, Oracle recommends that users use the command line to restart the OC4J instance.

3.1.7 TopLink Sessions Not Available in Application Server Control Console

If the TopLink Sessions for a TopLink-enabled application are not available in Application Server Control Console, check to be sure the TopLink session is configured to create the MBeans at login time. This is done by ensuring that the application has a `serverPlatform` class defined, and that the `ServerPlatform` class has its is `RuntimeServicesEnabled` flag enabled.

For Oracle Application Server 10*g* Release 3 (10.1.3), you should be using the following platform class, which can be set in the `sessions.xml` or through the session API:

`oracle.toplink.platform.server.oc4j.Oc4j_10_1_3_Platform`

When developing a TopLink-enabled application using Oracle JDeveloper, make sure to use version 11 or higher.

> **See Also:**   "Configuring the Server Platform" in the *Oracle TopLink Developer's Guide*

3.1.8 Unable to Receive MBean Notification Using OPMN to Start or Stop OC4J

You will not be able to receive notification from the `ias:j2eeType=J2EEServer,name...` MBean entity if you start or stop Oracle Containers for J2EE (OC4J) using OPMN. This happens using either the Application Server Control or the `opmnctl stop` or `opmnct start` command from the command line.

There is presently no workaround for this issue.

3.1.9 Using the Java Server Pages Standard Tag Libraries

The Java Server Pages Standard Tag Library (JSTL) makes use of Jaxp 1.2 classes that are packaged with Java Developer Kit 1.4.

Oracle Application Server 10*g* Release 3 (10.1.3) makes use of JDK 1.5 which uses Jaxp 1.3 classes. However, the JSTL still requires the Jaxp1.2 classes. If you run the JSTL with XML related tags in JDK 1.5 you may receive an error message similar to:

`: missing class org.apache.xpath.encounter failure.`

To avoid JSTL failure, include the `xalan.jar` file in the required `.war` file. Add the `xalan.jar` file into your `/WEB-INF/lib` directory with the `.war` file and then re-package.

For more information refer to the JSTL release notes at:

`http://java.sun.com/webservices/docs/1.6/jstl/ReleaseNotes.html`.

3.1.10 Error While Generating Web Service

When generating a Web Service from a stored procedure including XSL transformation, and returned data contain non-English characters, an error will be thrown as follows:

```
java.sql.SQLException: Invalid UTF8 encoding.
```

To workaround around this issue, in the `oracle/j2ee/ws/tools/wsa/db/webservices10literal.properties` file of `ORACLE_HOME/webservices/lib/wsa.jar` home, modify:

```
org.w3c.dom.Document _tmpDocument_ =db.parse(%1.getclobval().getAsciiStream());
```

to:

```
db.parse(new org.xml.sax.InputSource(__jRt_0.getclobval().getCharacterStream()));
```

3.1.11 Problem with Deployment of non-English Character Java Server Pages

When you deploy an application with non-English page character set Java Server Pages (JSPs) using a Web browser, accessing the JSPs leads to corresponding modification to the encoding setting of the Java-compiler. However, if you deploy applications which use the Java-compiler to compile auto-generated codes containing non-English characters (for example, an CMP application containing non-English characters as column names), an exception will be thrown as follows:

```
@ com.evermind.compiler.CompilationException: Syntax error in source or
compilation failed.
```

To workaround this problem:

1. Shutdown OC4J

2. Remove `encoding="<SOME-ENCODING>"` from the `<java-compiler>` element in the `ORACLE_HOME/j2ee/home/comfig/server.xml` file.

3. Restart OC4J.

3.1.12 Additional Step Required When Configuring ODL Logging for Oracle HTTP Server

For Oracle Application Server 10g Release 3 (10.1.3.0.0), if you follow the instructions in section 5.4.3.1, "Configuring Oracle HTTP Server to Produce ODL Messages" in the Oracle Application Server Administrator's Guide, you must perform the following additional task in order for the Oracle HTTP Server diagnostic log (`log.xml`) to show up in the Application Server Control Log Viewer:

1. Locate the following configuration file in the Oracle home where the Application Server Control (`ascontrol`) is deployed:

```
ORACLE_HOME/j2ee/home/applications/ascontrol/ascontrol/WEB-INF/config
```

```
/registration/OHS.xml
```

2. Edit the `OHS.xml` file and locate the `</logs>` tag at the end of the file.

3. Enter the following content immediately before the `</logs>` tag:

```
<log path="Apache/Apache/logs/oracle" componentId="OHS">
   <logreader type="ODL"/>
   <logviewer LogType='ERROR' ComponentType='HTTP Server' ComponentName='
    ID_VLOGS_VIEW_DIAG_LOGS@oracle.sysman.ias.studio.logging.msg.LoggingMsg'>
       <property name='category' value='diagnostic'/>
   </logviewer>
</log>
```

4. Restart the middle-tier instance.

   You can either restart the OC4J instance where Application Server Control is deployed or press the Page Refresh .gif file on the Log Files page.

3.2 Clustering and Replication Issues

This section describes clustering and replication issues. It includes the following topics:

- Section 3.2.1, "State Replication Framework"
- Section 3.2.2, "Using Oracle Universal Installer Provided Sample Cluster Discovery Address May Inadvertently Cluster Servers"
- Section 3.2.3, "Configuration of Oracle Application Server Clusters"

3.2.1 State Replication Framework

The state replication framework that is used by Oracle Application Server has been upgraded with several fixes that resolve multicast state replication issues and enable redeployment of running applications that use the framework.

Before deploying applications using the state replication framework in a production environment, go to Oracle*MetaLink* (`https://metalink.oracle.com`) and download and apply the required patch for Bug 4685049.

3.2.2 Using Oracle Universal Installer Provided Sample Cluster Discovery Address May Inadvertently Cluster Servers

Oracle Universal Installer provides an example cluster discovery address as part of the advanced installation option. The provide example discovery address is `225.0.0.1:6789`. This *is not* a recommended address; rather it is an example intended to provide the type of cluster discovery address users may ask for from their network administrator.

Because the cluster configuration of Oracle Application Server is fully dynamic it is possible for installations using the example cluster discovery address (`225.0.0.1:6789`) to be inadvertently clustered with other servers installed with the same example cluster discovery address.

The cluster discovery address of a specific Oracle Application Server instance can be set from the command line using the following `opmnctl` command:

```
> $ORACLE_HOME/opmn/bin/opmnctl config topology update discover=<cluster config
address>
```

For example, to update a cluster discovery address in a specific Oracle Application Server instance to be `225.0.0.1:9876`, the command would be:

```
> $ORACLE_HOME/opmn/bin/opmnctl config topology update discover="*225.0.0.1:9876"
```

Details on configuring topologies and the cluster discovery address can be found in Chapter 8, "Configuring and Managing Clusters" of the *Oracle Containers for J2EE Configuration and Administration Guide*.

3.2.3 Configuration of Oracle Application Server Clusters

Oracle Application Server instances can be grouped together in clusters using the *ORACLE_HOME*/bin/opmnassociate command line utility or explicitly using the more comprehensive *ORACLE_HOME*/opmnctl command line tool.

Additional information on post-installation topology and cluster configuration can be found in Chapter 8, "Configuring and Managing Clusters"" of the *Oracle Containers for J2EE Configuration and Administration Guide*.

3.3 Documentation Errata

The section describes documentation errata in management documentation. It includes the following topics:

- Section 3.3.1, "Incorrect Recommendation for opmn.xml File Edit"
- Section 3.3.2, "Incorrect Default Value for start-mode Element in Oracle HTTP Server opmn.xml file"
- Section 3.3.3, "Incorrect Remote Security Cross Reference"
- Section 3.3.4, ""retry" Attribute Not Valid for <ping> Element"
- Section 3.3.5, "Default Ping Timeout Value in OPMN Is 30 Seconds, Not 20"
- Section 3.3.6, "Incorrect Note for AggreSpy in Performance Guide"
- Section 3.3.7, "AggreSpy Security Information Update"
- Section 3.3.8, "Incorrect Ports Listed in the Oracle Application Server Administrator's Guide"
- Section 3.3.9, "Enabling and Disabling Components Is Supported"
- Section 3.3.10, "Incorrect request Attribute Description"
- Section 3.3.11, "Incorrect Title"

3.3.1 Incorrect Recommendation for opmn.xml File Edit

In Section 3.1, "opmn.xml", of the Oracle Process Manager and Notification Server Administrator's Guide the following is incorrectly stated:

"You can edit the `opmn.xml` file using Application Server Control Console. Click the **Process Management** link at the bottom of the Oracle Application Server instance home page. Do not stop the OPMN server after you edit the opmn.xml file. Application Server Control Console automatically reloads the updated opmn.xml file after you edit the file."

The Oracle Application Server installation does not contain a **Process Management** link at the bottom of the Oracle Application Server instance home page.

Therefore, you must perform modifications to the opmn.xml file using a text editing program.

3.3.2 Incorrect Default Value for start-mode Element in Oracle HTTP Server opmn.xml file

The id="start-mode" default value for the opmn.xml file listed in the Oracle HTTP Server chapter of the Oracle Process Manager and Notification Server Administrator's Guide is incorrectly listed as ssl-enabled. The default value for id="start-mode" is ssl-disabled.

3.3.3 Incorrect Remote Security Cross Reference

In Section 3.9.1, "Remote Security", of the Oracle Process Manager and Notification Server Administrator's Guide, there is an incorrect cross-reference to the Oracle Application Server Security Guide regarding more information about wallets, the orapki utility and other security issues. These references are incorrect. The Oracle Application Server Administrator's Guide contains information about security for the Oracle Application Server 10*g* Release 3 (10.1.3).

3.3.4 "retry" Attribute Not Valid for <ping> Element

Chapter 6, "opmn.xml Common Configuration", of the Oracle Process Manager and Notification Server Administrator's Guide lists retry as one of the attributes of the <ping> element. This is incorrect: the retry attribute cannot be used with the <ping> element.

You can set up a custom ping command in the opmn.xml file. For details, see chapter 10, "Configuring Custom Process", in the Oracle Process Manager and Notification Server Administrator's Guide.

3.3.5 Default Ping Timeout Value in OPMN Is 30 Seconds, Not 20

The *Oracle Process Manager and Notification Server Administrator's Guide* incorrectly states that the default ping timeout value is 20 seconds. It is actually 30 seconds.

3.3.6 Incorrect Note for AggreSpy in Performance Guide

In Section A.2.1, "Using the AggreSpy Display", of the *Oracle Application Server Performance Guide*, the note that mentions that "You can only run AggreSpy when the home OC4J instance is running", is incorrect and should be removed. The running OC4J instance can be any OC4J instance.

In the same note, the sentence, "Using an OracleAS Infrastructure, the home instance needs to be started to use AggreSpy, since by default the home instance is installed with OracleAS Infrastructure, but it is not started." should be removed.

3.3.7 AggreSpy Security Information Update

In Section A.2.3, "AggreSpy URL and Access Control", of the *Oracle Application Server Performance Guide*, the information on viewing metrics from a system other than the localhost is incomplete. To view metrics from other than the localhost, you also need to modify the /dmsoc4j location in the dms.conf file to allow access from a specific host, or from all hosts. By default, access is denied from all.

For example, to allow access from all hosts, modify dms.conf as follows:

```
<Location /dmsoc4j>
  SetHandler oc4j-handler
  Allow from all
</Location>
```

3.3.8 Incorrect Ports Listed in the Oracle Application Server Administrator's Guide

In Table D-1, "J2EE and Oracle HTTP Ports", in the Oracle Application Server Administrator's Guide, the allotted port range is incorrectly listed for the Listen (SSL) and SSL ports. The correct port numbers are as follows:

For the Listen (SSL) port:

- Allotted port range: 4443 - 4543

- Default port number: 4443

For the SSL port:

- Allotted port range: 4443 - 4543

- Default port number: 4443

3.3.9 Enabling and Disabling Components Is Supported

The *Oracle Application Server Administrator's Guide* incorrectly states that enabling and disabling components is not supported. Although it is not supported in Application Server Control Console, it is supported by OPMN. For more information on enabling and disabling components, see the *Oracle Process Manager and Notification Server Administrator's Guide*.

3.3.10 Incorrect request Attribute Description

In Oracle Process Manager and Notification Server Administrator's Guide, section 6.2, "opmn.xml Element and Attribute Descriptions", the `request` attribute description contains the following incorrect statement:

"The `request` attribute is for the IP address or host name to which ONS will bind its remote port."

The correct statement is:

"The `request` attribute is for the IP address or host name to which ONS will bind its request port."

3.3.11 Incorrect Title

In Section 7.6, "Generic Apache (Linux only)", of the Oracle Process Manager and Notification Server Administrator's Guide the title is incorrect. The listed configuration for the Oracle HTTP Server process module to manage generic Apache processes is valid for Linux and Microsoft Windows operating systems.

4

Oracle HTTP Server

This chapter describes issues associated with Oracle HTTP Server. It includes the following topics:

- Section 4.1, "Issues and Workarounds"
- Section 4.2, "Documentation Errata"

4.1 Issues and Workarounds

This section contains the following topics:

- Section 4.1.1, "Configuring Weighted Routing for AJP13 Destinations"
- Section 4.1.2, "Unable to Start Oracle HTTP Server for Turkish Language"
- Section 4.1.3, "Changing the Location of the PID File Requires a Change in apachectl"
- Section 4.1.4, "Routing Requests to Different Middle Tiers Based on the URL of the Request"

4.1.1 Configuring Weighted Routing for AJP13 Destinations

In the `Oc4jMount` directive, weighted load balancing works only when the destinations are instances or clusters. Weighted load balancing does not work for AJP13 destinations. For AJP13 destinations, the load is distributed evenly in a round-robin manner. For example, if your `mod_oc4j.conf` file contains the following lines, Host_A and Host_B will get an equal number of requests despite the settings in the `Oc4jRoutingWeight` directives.

```
Oc4jSelectMethod roundrobin:weighted
Oc4jRoutingWeight Host_A 1
Oc4jRoutingWeight Host_B 25
Oc4jMount /j2ee   ajp13://Host_A:<AJP Port>,Host_B:<AJP Port>
Oc4jMount /j2ee/* ajp13://Host_A:<AJP Port>,Host_B:<AJP Port>
# Instance weighted routing work as expected
#Oc4jMount /j2ee   instance://Host_A:home,Host_B:home
#Oc4jMount /j2ee/* instance://Host_A:home,Host_B:home
```

A possible workaround to achieve weighted load balancing for AJP13 destinations is to specify the same host multiple times in the `Oc4jMount` directive. The following example specifies Host_B twice.

```
Oc4jMount /j2ee ajp13://Host_A:<AJP Port>,Host_B:<AJP Port>,Host_B:<AJP Port>
```

4.1.2 Unable to Start Oracle HTTP Server for Turkish Language

When using the Turkish language with Oracle Application Server, you may see the following error:

```
Syntax erro on line 478 of /export/web/ohs/conf/httpd.conf:
Invalide command 'MIMEMagicFile', perhaps misspelled or defined in a module
not included in the server configuration.
```

To avoid this error, set the following directive:

```
LC_ALL=POSIX
```

4.1.3 Changing the Location of the PID File Requires a Change in apachectl

The `PidFile` directive specifies the location of the PID file. When Oracle HTTP Server starts up, it writes its process ID in the PID file.

If you edit the `PidFile` directive to change the location of the PID file, you must also make a corresponding change in the `ORACLE_HOME/Apache/Apache/bin/apachectl` file.

4.1.4 Routing Requests to Different Middle Tiers Based on the URL of the Request

It is not possible to configure two separate Oracle HTTP Servers using the same virtual hostname and port, where the proxy server routes the requests based upon the URL path, and also have them participate as partners in the same SSO configuration.

For example: You configure your load balancer or OracleAS Web Cache on the web tier to route requests to applications that use OracleAS Single Sign-On to the appropriate middle tier, based on the URLs.

- You have an Oracle HTTP Server configured to handle requests for `http://wwww.mycompany.com`.

- You have an OracleAS Web Cache configured to route requests for `http://www.mycompany.com/app1` to midtier1.

- You have an OracleAS Web Cache configured to route requests for `http://www.mycompany.com/app2` to midtier2.

`app1` and `app2` are applications that use OracleAS Single Sign-On.

When a user issues a request for `/app1`, the request is redirected to midtier1. mod_osso on midtier1 redirects the request to the SSO server for authentication. After authentication, the SSO server calls the success URL, which is always `www.mycompany.com/osso_login_success` regardless of which application the request was for, and this success URL is handled by the Oracle HTTP Server for `www.mycompany.com`. Because the success URL does not specify which application the request is for, the Oracle HTTP Server for `www.mycompany.com` does not know where to redirect the request (whether to redirect it to midtier1 or midtier2). The load balancer or OracleAS Web Cache cannot be configured to redirect the success URL correctly as there is no application information in the success URL.

A possible solution to this issue is described in MetaLink note 390358.1, which can be obtained from Oracle Customer Services via a Service Request.

4.2 Documentation Errata

This section describes documentation errata. It includes the following topics:

4.2.1 Default Values for Oc4jCacheSize

The "Understanding Modules" chapter of the *Oracle HTTP Server Administrator's Guide* contains the default value of 1 for Oc4jCacheSize.

The default value for `Oc4jCacheSize` should be 1.

4.2.2 UseOutputStreamSize

The "Using Oracle Containers for J2EE" appendix of the *Oracle HTTP Server Administrator's Guide* has a "Configuring OC4J Plug-in on Sun ONE" section that has the following example:

```
Service type="oracle/opii" fn="opii_service" UseOutStreamSize=8192
```

It should be:

```
Service type="oracle/opii" fn="opii_service" UseOutputStreamSize=8192
```

4.2.3 Correction to SSLCARevocationFile Directive Description

The description for the `SSLCARevocationFile` directive in *Oracle HTTP Server Administrator's Guide*, Chapter 10, "Enabling SSL for Oracle HTTP Server," should be corrected as follows:

Specifies the file where you can assemble the Certificate RevocationLists (CRLs) from CAs (Certificate Authorities) that you accept certificates from. These are used for client authentication. Such a file is the concatenation of various PEM-encoded CRL files in order of preference. CRL files should be from a single issuer. Files specified by `SSLCARevocationFile` should not be hashed. There should be only one `SSLCARevocationFile` entry; if there are multiple entries, then the last one will be used. `SSLCARevocationFile` can be used alternatively and/or additionally to `SSLCARevocationPath`.

4.2.4 Correction to SSLCARevocationPath Directive Description

The description for the `SSLCARevocationPath` directive in *Oracle HTTP Server Administrator's Guide*, Chapter 10, "Enabling SSL for Oracle HTTP Server," should be corrected as follows:

Specifies the directory where PEM-encoded Certificate Revocation Lists (CRLs) are stored. These CRLs come from the CAs (Certificate Authorities) that you accept

certificates from. If a client attempts to authenticate itself with a certificate that is on one of these CRLs, then the certificate is revoked and the client cannot authenticate itself with your server.

CRL files in the `SSLCARevocationPath` directory must be hashed. You can find the instructions to hash a CRL in *Oracle Application Server Administrator's Guide*, Section 12.2.5.2.1, "Renaming CRLs with a Hash Value for Certificate Validation." Note that `orapki` creates a file with a ".rN" extension. `SSLCARevocationPath` will not work with this extension and it is still possible to access with a revoked certificate. To get it to work with Oracle HTTP Server, change the extension from ".rN" to ".r0".

`SSLCARevocationPath` can be used alternatively and/or additionally to `SSLCARevocationFile`.

4.2.5 Oracle HTTP Server Apache Version Number

Oracle HTTP Server is based on Apache version 1.3.34.

4.2.6 Log Level Choices for Configuring IIS Listener for Single Sign-On are Incorrect

The "Configuring IIS Listener for Single Sign-On" section in *Oracle HTTP Server Administrator's Guide* incorrectly states that the valid log levels are `debug`, `inform`, `error`, and `emergency`.

The valid log levels are `debug`, `inform`, `error`, and `emerg`.

4.2.7 Incorrect Tags Listed for 40-Bit and 56-Bit Export Ciphers

Table 10-1, "SSLCipher Suite Tags", in the *Oracle HTTP Server Administrator's Guide* listed incorrectly the aliases for the 40-bit and the 56-bit export ciphers.

For 40-bit export cipher, do not use `EXP40`. Use `EXPORT40` instead.

For 56-bit export cipher, do not use `EXP56`. Use `EXPORT56` instead.

4.2.8 Incorrect Web Address for mod_php Extensions Information

The Web site provided for additional information on mod_php extensions was incorrect. The correct Web site is

http://www.php.net/manual/en/funcref.php

4.2.9 Clarification for the Name of the Oracle Application Server Proxy Plug-In Definition File

The "Using Oracle Application Server Proxy Plug-In" appendix in the *Oracle HTTP Server Administrator's Guide* mentions the proxy definition file without mentioning the actual filename. This is because the definition file can have any name.

If you are using the Oracle Application Server Proxy Plug-In with Microsoft IIS, you specify the full path to the definition file in the `server_defs` entry in the Windows registry. See the "Using Oracle Application Server Proxy Plug-In" appendix in the *Oracle HTTP Server Administrator's Guide* for the specific location in the registry.

If you are using the Oracle Application Server Proxy Plug-In with Sun Java System listener, you specify the full path to the definition file using the `server_defs` parameter on the `Init` line in the `magnus.conf` (or `obj.conf`, depending on the version of your Sun Java System listener) configuration file. In the following example, /oracle/proxyplugin/proxydefs is the definition file:

```
Init fn="op_init" server_defs="/oracle/proxyplugin/proxydefs" log_
file="/oracle/proxyplugin/oproxy.log" log_level=error
```

5

Oracle Containers for J2EE

This chapter discusses release notes for Oracle Containers for J2EE (OC4J) for 10.1.3. It includes the following topics:

- Section 5.1, "Configuration, Deployment, and Administration"
- Section 5.2, "Servlets and JSPs"
- Section 5.3, "EJB"
- Section 5.4, "Web Services"
- Section 5.5, "Web Services Security"
- Section 5.6, "OC4J Services"
- Section 5.7, "J2EE Connector Architecture (J2CA)"
- Section 5.8, "OracleAS JAAS Provider and Security"
- Section 5.9, "Documentation Errata"
- Section 5.10, "Oracle Application Server Containers for J2EE Job Scheduler"

You can access Oracle manuals mentioned in this document at the following URL:

http://www.oracle.com/technology/index.html

5.1 Configuration, Deployment, and Administration

This section describes configuration, deployment, and administration issues for Oracle Application Server Containers for J2EE (OC4J). This section covers the following topic(s):

- Section 5.1.1, "Remove the Tomcat Examples"
- Section 5.1.2, "Huge JSPs Can Fail with JDK 5.0 on AIX"
- Section 5.1.3, "Deprecated Environment Variables dedicated.connection, dedicated.rmicontext, and LoadBalanceOnLookup"
- Section 5.1.4, "Deprecated Environment Variable ejb.batch.compile"
- Section 5.1.5, "Deprecated orion-ejb-jar.xml Attributes"
- Section 5.1.6, "Web-Site-Related Options No Longer Available"
- Section 5.1.7, "Unsupported Methods in JMX MBeanServer and MBeanServerConnection Interfaces"
- Section 5.1.8, "Upgrade to Latest J2SE Release"

For information on configuring OC4J, see the Configuration Guide for OC4J at:

http://www.oracle.com/technology/index.html

5.1.1 Remove the Tomcat Examples

OC4J ships the Tomcat examples by default. Many of the Tomcat examples do not conform to the Oracle Secure Coding Standards. Oracle recommends that you remove the Tomcat examples except when used in demonstration and testing environments.

5.1.2 Huge JSPs Can Fail with JDK 5.0 on AIX

Using JSPs with a very large _jspService() method or mutiple try/catch/finally blocks, which generate many JSRs, can cause JSP compilation errors such as the following:

```
oracle.classloader.util.AnnotatedLinkageError: (_ias/_cluster/_topology)
```

```
jsr inlining encountered too many jsrs in method 1
(_jspService(Ljavax/servlet/http/HttpServletRequest;javax/servlet/http/HttpSer
vletResponse;)V) at pc=65535
```

Possible workarounds include a tag_reuse_default setting of "none" and usage of the reduceServiceMethodSize() method, or restricting yourself to JDK 1.4.2. A permanent solution will have to come from IBM (refer to PMR #08680,756)

5.1.3 Deprecated Environment Variables dedicated.connection, dedicated.rmicontext, and LoadBalanceOnLookup

Environment variables `dedicated.connection`, `dedicated.rmicontext`, and `LoadBalanceOnLookup` are deprecated.

To configure replication-based load balancing, use environment variable `oracle.j2ee.rmi.loadBalance` with the settings that Table 5–1 lists.

Table 5–1   Settings for Environment Variable oracle.j2ee.rmi.loadBalance

Setting	Description
client	The client interacts with the OC4J process that was initially chosen at the first lookup for the entire conversation (Default).
context	The client goes to a new server when a separate context is used (similar to deprecated `dedicated.rmicontext`).
lookup	The client goes to a new server for every lookup.

5.1.4 Deprecated Environment Variable ejb.batch.compile

Environment variable `ejb.batch.compile` is deprecated.

To enable or disable batch compilation, use the `orion-application.xml` file `<orion-application>` element `batch-compile` attribute.

5.1.5 Deprecated orion-ejb-jar.xml Attributes

The following `orion-ejb-jar.xml` file attributes are deprecated:

- `max-instances-per-pk`

- `min-instances-per-pk`

- `disable-wrapper-cache`

- `instance-cache-timeout`

- `locking-mode="old_pessimistic"`

> **Note:** Do not use these attributes in this release. Doing so will lead to deployment failure.

5.1.6 Web-Site-Related Options No Longer Available

The OC4J web-site-related options (accessible with the -site command) that were provided in the admin.jar utility in previous releases are no longer available.

For information on how to create and manage OC4J web-site configurations see the"Managing Web Sites in OC4J" chapter in the *Oracle Containers for J2EE Configuration and Administration Guide* .

5.1.7 Unsupported Methods in JMX MBeanServer and MBeanServerConnection Interfaces

A number of methods from the JMX MBeanServer interface are not available to J2EE applications when they use the MBeanServer object obtained from the following operation:

```
MBeanServer mbsrv = MBeanServerFactory.newMBeanServer();
```

The use of any of the following methods on the returned MBeanServer object will throw an UnsupportedOperationException exception:

```
public final ClassLoader getClassLoaderFor(ObjectName mbeanName)

public final ClassLoader getClassLoader(ObjectName loaderName)

public final ClassLoaderRepository getClassLoaderRepository()

public final Object instantiate(String className)

public final Object instantiate(String className, ObjectName loaderName)

public final Object instantiate(String className, Object[] params, String[]
signature)

public final Object instantiate(String className, ObjectName loaderName, Object[]
params, String[] signature)

public final ObjectInstance createMBean(String className, ObjectName name)

public final ObjectInstance createMBean(String className, ObjectName name,
ObjectName loaderName)

public final ObjectInstance createMBean(String className, ObjectName name,
Object[] params, String[] signature)

public final ObjectInstance createMBean(String className, ObjectName name,
ObjectName loader, Object[] params, String[] signature)

public final ObjectInputStream deserialize(ObjectName name, byte[] data)

public final ObjectInputStream deserialize(String className, byte[] data)

public final ObjectInputStream deserialize(String className, ObjectName
loaderName, byte[] data)
```

A number of methods from the MBeanServerConnection interface are not supported when an application uses the Oracle JMX connectors. The use of any of the

following methods on the `MBeanServerConnection` object that is created will throw an `UnsupportedOperationException` exception:

```
public final ObjectInstance createMBean(String className, ObjectName name)

public final ObjectInstance createMBean(String className, ObjectName name,
ObjectName loaderName)

public final ObjectInstance createMBean(String className, ObjectName name,
Object[] params, String[] signature)

public final ObjectInstance createMBean(String className, ObjectName name,
ObjectName loader, Object[] params, String[] signature)
```

5.1.8 Upgrade to Latest J2SE Release

Currently Oracle Application Server 10.1.3.0.0 is certified with AIX 1.4.2 (build ca142ifx-20050119) and AIX JDK 1.5.0 (build pap32dev-20051104). The product installs with AIX JDK 1.5.0 (build pap32dev-20051104) by default.

In general, J2SE releases are number *a.b.c_d*, where "*a.b.c*" is the major release number, as in 1.4.2 or 1.5.0, and "*d*" is the minor release number, as in "05" or "06". As a general practice, Oracle recommends that customers upgrade to the latest minor release number of J2SE to ensure that they benefit from any bugs resolved in those specific J2SE upgrades. Oracle explicitly restates the certification matrix for major release numbers of J2SE.

Currently there is a known J2SE bug in J2SE 1.5.0_ and J2SE 1.5.0_06 that manifests itself in an out-of-memory error in long-running stress tests involving `BigDecimals` numeric types. This bug is tracked by Sun at:

http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6360541
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6372116

The workaround for this bug is to upgrade to J2SE 1.5.0_06 and set the JVM startup parameters for the impacted Oracle Containers for J2EE instance with this additional parameter:

```
-XX:CompileCommand=exclude,oracle/jdbc/driver/NumberCommonaccessor.getBigdecim
```

Information on configuring the J2SE runtime in Oracle Application Server can be found in the *Oracle Containers for J2EE Configuration and Administration Guide* at:

http://www.oracle.com/technology/index.html

5.1.9 Workaround for ORA-604/ORA-12705 Error Using a Not-Fully Supported Locale

When you try to get a connection on a Locale that is not supported in JDBC, JDBC throws a SQLException. - Bug 4704421

Use the following to verify runtime Java's locale:

```
System.out.println(Locale.getDefault().toString())
```

Unsupported Locales include any Locale that is NOT listed in the "Fully Supported Locales" table in on the Java 5.0 Java Supported Locales page at the following URL: http://java.sun.com/j2se/1.5.0/docs/guide/intl/locale.doc.html

For example, Locales that are "Provided but not Tested" include the following:

- ab_CD

- fr_FR_EURO

- it_IT_EURO

- th_TH_TH, Thai (Thailand,TH)

- be, Belarusian

- be_BY, Belarusian (Belarus)

- es_AR, Spanish (Argentina)

- es_BO, Spanish (Bolivia)

- es_DO, Spanish (Dominican Republic)

- es_EC, Spanish (Ecuador)

- es_HN, Spanish (Honduras)

- es_PY, Spanish (Paraguay)

- es_UY, Spanish (Uruguay)

- mk, Macedonian

- mk_MK, Macedonian (Macedonia)

- no_NO_NY, Norwegian (Norway,Nynorsk)

- sq, Albanian

- sq_AL, Albanian (Albania)

The workaround for this problem is to update your default Locale setting of Java. You can do any of the following:

- Change default Locale from the unsupported `es_AR`, to the fully supported `es` (Spanish):

  ```
   Locale.setDefault(new Locale("es"));
  ```

- When Locale has variant code such as `fr_FR_EURO`, remove variant code (EURO) and set default:

  ```
  Locale.setDefault(new Locale("fr", "FR"));
  ```

- Set English as the default Locale:

  ```
  Locale.setDefault(Locale.ENGLISH);
  ```

5.1.10 Incompatibility When Moving Between JDK 1.5 and 1.4

When you deploy an application (including the OC4J default application) to OC4J running JDK 1.5 (Java 5), you cannot re-use that deployment on OC4J running JDK 1.4.

Code compiled with JDK 1.5 (Java 5) cannot be read by the JDK 1.4 VM. When OC4J is running under JDK 1.4 and tries to load a class which was compiled with JDK 1.5, a class loading exception will be thrown with the following message:

```
Unsupported major.minor version 49.0
```

This can occur in scenarios such as:

- You deploy an application that contains EJBs to OC4J running under JDK 1.5, then, without undeploying the application, you restart OC4J under JDK 1.4. The problem is that the generated code associated with the EJBs will have been

compiled with the same JDK version that was used to start the server and that the generated code is cached between server restarts on the file system in the *OC4J_HOME*/j2ee/home/application-deployments directory.

The workaround for this is to shutdown the server, remove either the contents of the *OC4J_HOME*/j2ee/home/application-deployments directory (or just the offending application's sub-directory) and restart the server with JDK 1.4.

- You deploy an EAR file which contains classes that were compiled with and targeted to JDK 1.5 to OC4J running under JDK 1.4.

  The workaround for this is to recompile the contents of the EAR using JDK 1.4 and redeploy.

> **Note:** To simplify this discussion, we assume that no cross compilations are being used to target code to specific JDK versions.

5.1.11 Configuring a Machine to Work With and Without a Network Connection

When you work on a single machine using localhost, add the IP address in the <ipaddr> subelement of the <notification-server> element and explicitly set up a discover list in the <discover> element to refer to the localhost OPMN remote port, as defined in the cluster <port> element. An example of this configuration follows:

```
<notification-server>
    <ipaddr remote="127.0.0.1" request="127.0.0.1"/>
    <port local="6101" remote="6201" request="6004"/>
    <ssl enabled="true"
wallet-file="$ORACLE_HOME\opmn\conf\ssl.wlt\default"/>
    <topology>
        <discover list="localhost:6201"/>
    </topology>
</notification-server>
```

If you supply the localhost IP address, 127.0.0.1, the machine can work with or without a network.

5.1.12 Converting Pre-10.1.3 Data Sources to 10.1.3 Format

For information on converting pre-10.1.3 data sources to 10.1.3 format, see the release note at Section 5.6.3.3, "Converting Existing Data Sources to Release 3 Format".

5.1.13 Xalan Library Not Supported as a Shared Library with JDK1.4

If you are using JDK1.4, Oracle Application Server 10.1.3 does not support using the Xalan library shipped with the JDK as a shared library. To use the Xalan library, you have two alternatives:

- Use JDK1.5, in which the embedded Xalan library IS supported as a shared library.

- With JDK1.4, use a standalone distribution of the Xalan library instead of the embedded version.

5.1.14 Recommendation for <cluster> Element write-quota Setting

Chapter 9 of the 10.1.3 *Oracle Containers for J2EE Configuration and Administration Guide* documents the <cluster> element of the orion-application.xml file, including

use of the write-quota attribute. This attribute determines the number of other "group members" within a cluster to which application state information should be replicated.

Be aware, however, that a "group member" is actually a JVM, not a node, and that it is possible to have multiple JVMs per node.

To ensure that more than one node receives state replication, set write-quota to a number greater than the highest number of JVMs on any one node within the cluster. For example, if there are three nodes, which have six JVMs, four JVMs, and three JVMS, respectively, set write-quota to a value of at least 7.

5.1.15 Configuring OC4J Instance for OCI Drivers

If any application uses oci drivers for a datasource connection then add the following in opmn.xml for that particular OC4J instance. For example, if the application is deployed to the default home instance, then configure the OC4J instance as follows:

```
<ias-component id="OC4J">
<environment>
<variable id="LIBPATH" value="$ORACLE_HOME/lib32" />
</environment>
<process-type id="home" module-id="OC4J" status="enabled">
```

5.1.16 The oracle.home and oracle. j2ee.home System Properties

The following information has been added to the 10.1.3.1 Oracle Containers for J2EE Configuration and Administration Guide.

Instead of the environment variables ORACLE_HOME and J2EE_HOME, you can use the system properties oracle.home and oracle.j2ee.home to set the Oracle and J2EE home directories. The following table describes these system properties that you can set for OC4J.

Property	Description
oracle.home	Sets the root directory into which you will install the OC4J distribution.
	Instead of using the oracle.home system property, you can set the value of the ORACLE_HOME environment variable to the root directory.
oracle.j2ee.home	Sets the J2EE home directory to the installed directory of the oc4j.jar and admin.jar files, *oc4j_install_dir*/j2ee/home.
	Setting this system property or the J2EE_HOME environment variable to the J2EE home directory enables you to invoke oc4j.jar and admin.jar from any directory.

5.1.17 Maximum Threads in Work Manager Thread Pool

The max attribute of the <work-manager-thread-pool> element or WorkManagerThreadPool MBean specifies the maximum number of threads that OC4J can create in the pool. The default value of max is 40.

The work manager thread pool uses three worker threads for internal use. For example, if you specify `max="16"`, then only 13 worker threads are available to service requests. Similarly, if the `max` value is `20`, then only 17 threads are available. So you need to set this value to your required maximum number of threads plus 3.

5.1.18 Configuring Multicast Discovery with opmnassociate

The `opmnassociate` utility adds the default `home` OC4J instance to a cluster using multicast discovery. This utility performs the following steps:

1. Inserts or updates the `<discover>` element in `opmn.xml` with the specified multicast address and port

2. Configures the `default` Web site to receive and respond to requests from Oracle HTTP Server using the Apache JServ Protocol (AJP) by modifying the corresponding `<port>` element in `opmn.xml`

3. Restarts OPMN to load the new configuration into the runtime

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

The syntax is as follows:

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

For example:

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

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

5.1.19 Downgraded JDK and <java-compiler> in server.xml

An OPMN-managed OC4J instance installed as a component of Oracle Application Server uses JDK 5.0 by default. If an application that will be deployed to OC4J requires a JDK 1.4.2 release, you can *downgrade* to the earlier version.

1. Stop the OC4J instance.

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

   Deleting this directory will cause the application files to be recompiled when OC4J is restarted with the JDK 1.4.2.

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

You can specify the JDK to use for each OC4J instance through manual edits to the `opmn.xml` configuration file. If you want to use the `javac` compiler installed with the JDK defined in the `JAVA_HOME` environment variable, also remove the `<java-compiler>` element from `server.xml` and let OC4J rediscover the default settings.

5.1.20 Configuration of Application Mount Points

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

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

You can still configure application mount points manually and view the mount point list, as the *Oracle Containers for J2EE Configuration and Administration Guide* describes. For additional information about configuring mount points, see *Oracle HTTP Server Administrator's Guide*.

5.1.21 Hot Deployment in OC4J

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

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.

For details on redeploying applications to an OC4J instance, see the *Oracle Containers for J2EE Deployment Guide*.

5.1.21.1 Deployment of Applications

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.

5.1.21.2 Deployment of Shared Libraries

Shared libraries are loaded and managed at a container level. A change to a shared library does require a container restart.

If a redeployed application has an `import-shared-library` declaration that goes from a lower to a higher version, such as from 2.2.8 to 2.2.9, then the version change in the declaration should not require a container restart. The application will be restarted on redeployment and should pick up the new shared library version (as long as the library is available).

5.1.22 Allowed Format for Version Numbers in MANIFEST.MF File

Previously, version numbers in the `Specification-Version` and `Implementation-Version` attributes of a `MANIFEST.MF` file were limited to five elements: `n1.n2.n3.n4.n5`. (OC4J would not start if there were more than five elements in a `Specification-Version` or `Implementation-Version` value.)

Now, version numbers of up to eight elements are allowed: `n1.n2.n3.n4.n5.n6.n7.n8`.

The maximum allowed value for an element is `99999999`.

5.1.23 Migration Tool Issue: Prepending Oracle Internet Directory Realm Name to Custom Principals

Be aware of the following issue when you use the OracleAS JAAS Provider migration tool to migrate policies from the file-based provider to the Oracle Identity Management (essentially, Oracle Internet Directory) security provider, either in "policy" mode or "all" mode: The migration tool prepends the Oracle Internet Directory realm name to custom or nonrealm principal names in grantee entries in the

policy configuration. (A custom principal may come into play when authenticating through a custom login module, for example.)

In the migrated configuration, a custom principal name in a grantee entry becomes, for example, "us/anyone" instead of just "anyone", assuming "us" is the realm name. This results in permission issues. For ADF applications, for example, this results in public pages not working after migration to Oracle Internet Directory as the security provider.

This problem will be fixed in the 10.1.3.3 release. For 10.1.3.2 and prior 10.1.3.x releases, you can use either of the following workarounds:

- Manually remove the "us/" prefix from the LDIF file that the migration tool creates, prior to importing it into Oracle Internet Directory.

- Manually remove the "us/" prefix from relevant grantee entries after the migration, using Oracle Internet Directory administration tools.

5.1.24 Job Scheduler Redeployment

If you redeploy an application with scheduled jobs, the scheduled jobs will not run after the application is redeployed.

When redeploying an application with scheduled jobs, you should:

1. Remove all scheduled jobs.

2. Redeploy the application.

3. Resubmit all the jobs.

5.1.25 Incompatible ONS Versions

The Oracle Notification Service (ONS) that is included with the Oracle Application Server 10.1.3.x releases is incompatible with the ONS version included with the Database 10.1.0.x releases. The following error is received when the two versions attempt a connection:

```
invalid connect server IP format
...
Terminating connection
...
```

The incompatibility causes RAC features, such as Fast Connection Failover, to fail. The workaround is to install the 10.1.0.6 patch set for the Oracle Database Server. The patch set includes an updated ONS. The patch can be downloaded from https://metalink.oracle.com.

5.1.26 Use of ’/’ for Context Root

Specification of "/" for the context root when deploying an application to OC4J is supported now. This includes support by Application Server Control and admin_client.jar.

Background: The 10.1.3.1 release of the *Oracle Containers for J2EE Configuration and Administration Guide* states that "Specifying a root setting of "/" will override the OC4J default Web application. This setting or a null setting is not allowed by the admin_client.jar utility when binding a Web application to the Web site."

Now, however, a `root` setting of "/" *is* allowed. You can use this as the context root when deploying an application. The following example uses `admin_client.jar` to deploy a WAR file and bind to "/".

```
% java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1 \
  -deploy -file d:how-to-rolling-upgrade-web-v1.war -deploymentName h2ru_2 \
  -bindAllWebApps -contextRoot "/"
```

Be aware that if an EAR file includes an `application.xml` file that has the context root set to "/", such as in the following example, then "/" will be the default context root when the application is deployed using either Application Server Control or `admin_client.jar`.

```
<application>
  <display-name>How-To Rolling Upgrade</display-name>
  <module>
    <web>
      <web-uri>how-to-rolling-upgrade-web.war</web-uri>
      <context-root>/</context-root>
    </web>
  </module>
</application>
```

> **Note:** Because the default ping URL for Oracle HTTP Server is also "/", using "/" as the context root when you deploy an application may result in the following issues:
>
> - Pings intended for Oracle HTTP Server go directly to OC4J instead.
>
> - Extraneous HEAD requests appear in the `*-web-access.log` file.
>
> You can avoid these issues by placing the following directive:
>
> ```
> Oc4jMountCopy off
> ```
>
> Into the following file:
>
> ```
> ORACLE_HOME/Apache/Apache/conf/dms.conf
> ```

5.1.27 Warning Regarding Maximum Concurrent Timers

This release note provides background information and workarounds for warnings, such as the following examples, that occur when the number of concurrent timers exceeds the maximum.

```
WARNING J2EE OJR-10002

The number of concurrent Timers has reached the maximum limit
```

By default in 10.1.3.1 and 10.1.3.2, OC4J allows only eight concurrent timers. (A timer can be triggered through an EJB timer, the timer service, or the scheduler.) This limit is low by default because we expect each timer to be of short duration. When the number of timers is at the limit, such as if timers are running longer for any reason, timers are no longer executed; when a new timer occurs, you will see the warning messages. There are two OC4J flags you can use in this circumstance:

- `timer.service.debug`: Determines whether to output additional diagnostic information for the timer service, including information about the current number of running timers. For example: `-Dtimer.service.debug=true`

- `executor.concurrent.tasks`: Specifies the number of concurrent tasks for the Executor Service. Through this flag you can increase the maximum number of concurrent timers allowed by OC4J. For example: `-Dexecutor.concurrent.tasks=12`

> **Note:** Each timer executes in a separate thread. If the maximum is set too high, resulting in numerous timers executing, then many threads will be used. It is advisable to recycle threads once they finish executing.

5.1.28 Incorrect orion-ejb-jar.xml Created When cmp-field-mapping Specified

In previous releases of OC4J, when you use `ascontrol` to deploy an EJB CMP 2.1 application with an `orion-ejb-jar.xml` file that contains `cmp-field-mapping` elements, OC4J sometimes failed to preserve `persistence-name`, omitted `persistence-type`, or both.

In this release, OC4J correctly handles all `cmp-field-mapping` configuration.

5.1.29 Specifying the Mapping Attribute

For X.509 Certificate Authentication (X509Certificate), JAZN sets the mapping attribute for the Oracle Internet Directory (OID) provider to `"DN"` by default.

To change the default value, you can configure the value of the `mapping.attribute` property:

1. Locate the `$Oracle_Home/j2ee/<oc4j_inst>/config/jazn.xml` file.

2. Add the `mapping.attribute` property configuration to the <jazn> tag. For example:

```
<jazn provider... >
...
    <property name = "mapping.attribute" value="cn"/>
...
</jazn>
```

3. Restart the OC4J instance.

Likewise, other login modules now rely on the `mapping.attribute` property in the `jazn.xml` file, including:

- WS-Security Username token with no password (`WSSLoginModule`)

- SAML (`SAMLLoginModule`)

  `SAMLLoginModule` first looks at the `SubjectNameIdentifier` to determine the mapping. If blank, then the `mapping.attribute` is used.

- X.509 client certificate authentication (`X509LoginModule`)

- Third party login module (`LDAPLoginModule`)

The mapping attribute for these login modules must be set in the `jazn.xml` file as described here.

5.1.30 Use of the http.file.allowAlias Property

By default, OC4J now ships with the `http.file.allowAlias` property set to `false`. This setting prevents the use of symbolic links. Oracle strongly recommends that this setting not be changed to `true`, which might allow JSP source code to be visible to end users in some circumstances.

Instead of changing the property setting, you can use one of the following workarounds:

■ Temporarily switch from using the OC4J lightweight HTTP listener to front ending the OC4J application through Oracle HTTP Server, so that browsers access the pages indirectly through MOD_OC4J and Apache JServ Protocol (AJP), rather than directly through HTTP.

■ Replace all symbolic links in an application with the names of the real files they represent.

You can use a shell script to automate the replacement of symbolic links. For example:

```
#!/bin/ksh

PROGNAME="${0##*/}"
LN_EXTN="ln"

function displaySyntax
{
echo "${PROGNAME}! SYNTAX: ${PROGNAME} <some_dir_path>"
exit 1
}

if [[ $# < 0 ]]
then
displaySyntax
fi

DIR="$1"

if [[ ! -d ${DIR}
then
displaySyntax
fi

find ${DIR} -type l|while read filepath
do
echo "FIXING: ${filepath} (=> ${filepath}.${LN_EXTN})"
mv ${filepath} ${filepath}.${LN_EXTN}
cp -L ${filepath}.${LN_EXTN} ${filepath}
done
```

This example KSH script would be invoked on Unix as follows:

```
$ fixLinks <web_module_root>
```

The script will recurse through any directory and, for any file it finds that is a symbolic link, will rename each link with an additional `.ln` extension and then place a copy of the link target in the original location where the link was found.

5.1.31 oracle.http.client Shared Library Missing from system-application.xml

The `oracle.http.client` shared library is missing from the `system` application. This library, which should be part of the set of shared libraries that applications import from `system` by default, is not specified in the 10.1.3.0.0 `system-application.xml` configuration file.

Before you deploy any portlets to an OC4J instance, you need to add the `oracle.http.client` shared library to the *ORACLE_HOME*/j2ee/*instance*/config/system-application.xml file. You can do this by adding the following line to the `<imported-shared-libraries>` section of the file:

```
<import-shared-library name="oracle.http.client" />
```

5.2 Servlets and JSPs

This section describes release notes for servlets. It covers the following topic(s):

- Section 5.2.1, "Servlet Invocation by Classname Disabled by Default"

- Section 5.2.2, "Physical File Required for Welcome File"

- Section 5.2.3, "Warning Issued for servlet.init() Not Working with run-as"

- Section 5.2.4, "Request Parameters Not Available During Filter Execution"

- Section 5.2.5, "The default-charset Setting Does Not Apply to Servlets"

- Section 5.2.6, "Form Authentication Filter"

- Section 5.2.7, "ojsp Tag Libraries Deprecated"

- Section 5.2.8, "Disabling Servlet Reloading"

- Section 5.2.9, "Specifying justrun for Production Environment Efficiency"

- Section 5.2.10, "Using a Tag Library with the ojspc Utility"

5.2.1 Servlet Invocation by Classname Disabled by Default

In the 10.1.3 implementation, servlet invocation by class name is not enabled by default. Therefore, in default mode, you must use standard servlet configuration in `web.xml` before a servlet can be invoked. For example:

```
<servlet>
   <servlet-name>mytest</servlet-name>
   <servlet-class>mypackage.MyTestClass</servlet-class>
</servlet>
...
<servlet-mapping>
   <servlet-name>mytest</servlet-name>
   <url-pattern>/servlet/mytest</url-pattern>
</servlet-mapping>
```

Without this configuration, attempts to invoke the servlet will result in a `404 NOT FOUND` error. This differs from the default behavior in previous releases, where invocation by class name was enabled.

Alternatively, customers can choose to enable invocation by class name when they start OC4J, by setting the `http.webdir.enable` property as follows:

```
-Dhttp.webdir.enable=true
```

5.2.2 Physical File Required for Welcome File

A physical file must be present for a welcome file to dispatch to a servlet. To create a servlet mapped to /index.html that maps to the JSP /index.jsp and have it serve as a welcome file, the web.xml file should include the following entries:

```
<servlet>
   <servlet-name> index_jsp </servlet-name>
   <jsp-file> /index.jsp </jsp-file>
</servlet>

<servlet-mapping>
   <servlet-name>index_jsp</servlet-name>
   <url-pattern>/index.html</url-pattern>
</servlet-mapping>
```

This works *only* if there is a physical file, /index.html, in the Web application. The file can be zero length. As long as the file exists, this servlet will be loaded as the welcome file. Otherwise, a java.lang.StringIndexOutOfBoundsException exception will be thrown.

5.2.3 Warning Issued for servlet.init() Not Working with run-as

For a Web application, when run-as *user* is specified in the web.xml file, all method invocations except the Servlet.init() method will be invoked as the specified user. With the JMS Router being the default application of OC4J, calls need to be authorized to the router's EJBs. This is done by defining the application role "jmsRouter", which is mapped to the JAAS "oc4j-administrators" role, and specifying <method-permission> for all methods of the router's EJBs.

The init() method of the servlet within the router's Web model creates a router EJB object. Regardless of whether run-as is specified in web.xml for the servlet, a security exception is thrown:

```
@ oracle.oc4j.rmi.OracleRemoteException: anonymous is not allowed to call this EJB
method, check your security settings (method-permission in ejb-jar.xml and
security-role-mapping in orion-application.xml).
```

Workaround

The security warning can be removed by commenting out '*' in the <method-name> element of <method-permission> in ejb-jar.xml and explicitly enumerating all methods in AdminMgrBean that the jmsRouter role can access, as follows.

```
<!--
    <method-permission>
      <role-name>jmsRouter</role-name>
      <method>
         <ejb-name>AdminMgrBean</ejb-name>
         <method-name>*</method-name>
      </method>
    </method-permission>
-->
    <method-permission>
      <role-name>jmsRouter</role-name>
      <method>
```

```
    <ejb-name>AdminMgrBean</ejb-name>
    <method-name>getConfig</method-name>
  </method>
</method-permission>
...
```

runAsRoleName is correctly parsed in `ServletDescriptor.java`, stored in `info` and `thread` in `HttpApplication.loadServlet()`.

5.2.4 Request Parameters Not Available During Filter Execution

HTTP request parameters will not be available to servlet filters that are meant to be executed before dispatch of the request to a static resource (an `.html` file, for example). Note that filters that execute before dynamic resources, such as a servlet or JSP, will have access to the parameters.

5.2.5 The default-charset Setting Does Not Apply to Servlets

In 10.1.3, the `default-charset` attribute of the `<orion-web-app>` element in the `orion-web.xml` file applies to JSP pages only, not to servlets.

5.2.6 Form Authentication Filter

An OC4J proprietary filter dispatcher was introduced in 10.1.3 that enables a filter to access the username/password credentials passed in to OC4J through Form Authentication. For example, you would do this if you want to perform further authentication on external resources.

To enable this feature, specify the FORMAUTH value in the `<dispatcher>` element for the filter in the `orion-web.xml` file.

The following examples show the code that declares the filter and the code that specifies the FORMAUTH feature:

Note:

The `<filter>` element can be declared in either the `orion-web.xml` file or the `web.xml` file.

The `<filter-mapping>` element must be declared in the `orion-web.xml` file.

In `orion-web.xml` or `web.xml`:

```
<filter>
   <filter-name>MyFilter</filter-name>
   <filter-class>myFilterClass</filter-class>
</filter>
```

In orion-web.xml:

```
<orion-web-app>
...
   <web-app>
   ...
      <filter-mapping>
         <filter-name>MyFilter</filter-name>
         <dispatcher>FORMAUTH</dispatcher>
      </filter-mapping>
```

```
    </web-app>
</orion-web-app>
```

Any filter declared this way will be executed after the authentication form is submitted, but before authentication is performed in OC4J.

The filter can call `request.getParameters()` to get the `j_username` and `j_password` parameters from the request object.

Here is an example of the calling code:

```
import javax.servlet.*;
import javax.servlet.http.*;

public class MyFilter implements Filter {

    public MyFilter() {
        super();
    }

    public void doFilter(ServletRequest request,
                         ServletResponse response,
                         FilterChain filterChain)
        throws IOException, ServletExxception {
        HttpServletRequest req = (HttpServletRequest) request;

String username = req.getParameter("j_username");
String password = req.getParameter("j_password");
// use these credentials to access a remote DB
....
filterChain.doFilter(request, response);
    }

    public void init(FilterConfig filterConfig) throws ServletException {
    }

    public void destroy() {
    }
}
```

5.2.7 ojsp Tag Libraries Deprecated

Oracle-proprietary ojsp tag libraries are deprecated in OC4J 10.1.3.x releases. They will be desupported in OC4J release 11g.

5.2.8 Disabling Servlet Reloading

In the `global-web-application.xml` file, you can force OC4J to not reload a specific servlet if the servlet is modified. This feature is enabled by setting the undocumented `auto-reload` attribute of the `<servlet>` element to `false`.

`auto-reload="false">`.

This parameter is not documented in the OC4J 10.1.3.x Servlet documentation.

The following example shows that this can be done for a single servlet.

In `global-web-application.xml`:

```
...
  <servlet auto-reload="false">
     <servlet-name>nlservlet</servlet-name>
       <servlet-class>com.netledger.core.requesthandler.NLServlet
       </servlet-class>
  </servlet>
```

5.2.9 Specifying justrun for Production Environment Efficiency

For efficiency in the production environment, the `sample-web.war` that is to be precompiled using `ojspc` should include the following declaration in its `sample-web.war\WEB-INF\web.xml` just after the `<web-app>` element:

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>oracle.jsp.runtimev2.JspServlet</servlet-class>
  <init-param>
     <param-name>main_mode</param-name>
     <param-value>justrun</param-value>
  </init-param>
</servlet>
```

The settings of the `main_mode` parameter, including `justrun`, are described in Chapter 3, Configuring the OC4J JSP Environment, of the *Oracle Containers for J2EE Support for JavaServer Pages Developer's Guide* .

5.2.10 Using a Tag Library with the ojspc Utility

To use your own tag library with `ojspc`, the best method is to put your tag library under `WEB-INF/lib` in a WAR that contains a JSP page and then precompile the archive, which can be either a standalone WAR or an EAR that contains the WAR.

For example, `mytaglib.war` contains these files:

```
jsp/mytaglib_example.jsp
WEB-INF/web.xml
WEB-INF/orion-web.xml
WEB-INF/lib/mytaglib.jar
```

The `mytaglib.jar` file contains `META-INF/mytaglib.tld` as well as the class files for all tags declared in that tag library descriptor (`mytaglib.tld`).

In this example, the WAR (`mytaglib.war`) is inside an EAR file along with `META-INF/application.xml`.

The following `ojspc` command precompiles the EAR file:

```
ojspc -output out/mytaglib.ear mytaglib.ear
```

After the precompilation, the EAR file will be in `out/mytaglib.ear` and will contain a WAR file with the precompiled JSP page under `WEB-INF/lib`, `__oracle_jsp_mytaglib_web.jar mytaglib.jar`, as well as the original `mytaglib.jar` file.

You could deploy this EAR file to OC4, and if you set the `<ojsp-init>` attribute `main-mode` to the value `justrun` in the `orion-web.xml` deployment descriptor, the

precompiled version in `WEB-INF/lib` will be used. Alternatively, you could set the `main_mode` parameter to `justrun` in an `<init-param>` element of the `jsp` servlet configuration for the OC4J instance in `$J2EE_HOME/config/global-web-applications.xml`.

5.3 EJB

This section describes release notes for EJB. It covers the following topics:

- Section 5.3.1, "EJB 3.0 Support"
- Section 5.3.2, "Orion CMP is Deprecated"
- Section 5.3.3, "Orion CMP and Non-Oracle Databases"
- Section 5.3.4, "Stateful Session Bean Replication Trigger Configuration"
- Section 5.3.5, "EJB 3.0 Entities and Application Server Control"
- Section 5.3.6, "Entity and Session Deployment Attribute tx-retry-wait"

5.3.1 EJB 3.0 Support

In this release, OC4J supports a subset of the functionality specified in the EJB 3.0 proposed final draft at
`http://jcp.org/aboutJava/communityprocess/pr/jsr220/index.html`

For example, support for some EJB 3.0 features such as persistence API, external lifecycle listener class, and interceptors may not be compliant with the latest EJB 3.0 specification.

You may need to make code changes to your EJB 3.0 OC4J application after the EJB 3.0 specification is finalized and OC4J is updated to full EJB 3.0 compliance.

5.3.2 Orion CMP is Deprecated

The Orion persistence manager is deprecated. Oracle recommends that you use OC4J and the TopLink persistence manager for new development. Using the migration tool, you can easily migrate an existing OC4J application that uses EJB 2.0 entity beans with the Orion persistence manager to use EJB 2.0 entity beans with the TopLink persistence manager.

For more information, see "Migrating OC4J Orion Persistence to OC4J TopLink Persistence" in the *Oracle TopLink Developer's Guide*.

5.3.3 Orion CMP and Non-Oracle Databases

When using the (deprecated) Orion persistence manager with CMP and a non-Oracle database, OC4J does not read the schema XML file specified by the `data-sources.xml` file `managed-data-source` element `schema` attribute.

For example, consider the `data-sources.xml` and `orion-ejb-jar.xml` files shown in the following examples:

Example 5–1   Non-Oracle Database data-sources.xml

```
<connection-pool
    name="ConnectionDB2"
    max-connections="20"
    min-connections="1">
    <connection-factory
```

```
                factory-class="com.oracle.ias.jdbcx.db2.DB2DataSource"
                user="jdoe"
                password="password"
                url="jdbc:oracle:db2://server.foo.com:50000;...>
                <property name="databaseName" value="appdb"/>
                <property name="packageName" value="JDBCPKG"/>
                <property name="serverName" value="server.foo.com"/>
                <property name="portNumber" value="50000"/>
                <xa-recovery-config>
                    <password-credential>
                        <username>jdoe</username>
                        <password>password</password>
                    </password-credential>
                </xa-recovery-config>
            </connection-factory>
        </connection-pool>

<managed-data-source
        connection-pool-name="ConnectionDB2"
        schema="database-schemas/db2.xml"
        jndi-name="jdbc/OracleDS"
        name="OracleDS"
/>
```

Example 5–2   orion-ejb-jar.xml

```
<enterprise-beans>
    <persistence-manager name="orion"/>
    <entity-deployment name="EmployeeBean" max-tx-retries="0" location="EmployeeBean">
        <primkey-mapping>
            <cmp-field-mapping
                name="empNo" persistence-name="empNo" persistence-type="integer"
            />
        </primkey-mapping>
        <cmp-field-mapping
            name="empName" persistence-name="empName"
        />
        <cmp-field-mapping
            name="salary" persistence-name="salary"
        />
        <finder-method lazy-loading="true">
            <method>
                <ejb-name>EmployeeBean</ejb-name>
                <method-name>findAll</method-name>
                <method-params></method-params>
            </method>
        </finder-method>
    </entity-deployment>
</enterprise-beans>
```

Deploying this application will raise an error like:

```
Error creating table: [oias][DB2 JDBC Driver][DB2]ILLEGAL SYMBOL
```

To work around this problem, update the orion-ejb-jar.xml to manually define the mapping data types as Example 5–3 shows.

Example 5–3   Updated orion-ejb-jar.xml

```
<enterprise-beans>
    <persistence-manager name="orion"/>
    <entity-deployment name="EmployeeBean" max-tx-retries="0" location="EmployeeBean">
        <primkey-mapping>
            <cmp-field-mapping
                name="empNo" persistence-name="empNo" persistence-type="integer"
```

```
                    />
               </primkey-mapping>
               <cmp-field-mapping
                   name="empName" persistence-name="empName" persistence-type="varchar(255)"
                />
               <cmp-field-mapping
                   name="salary" persistence-name="salary" persistence-type="double"
               />
               <finder-method lazy-loading="true">
                   <method>
                       <ejb-name>EmployeeBean</ejb-name>
                       <method-name>findAll</method-name>
                       <method-params></method-params>
                   </method>
               </finder-method>
           </entity-deployment>
   </enterprise-beans>
```

5.3.4 Stateful Session Bean Replication Trigger Configuration

In this release, for stateful session beans, OC4J supports `session-deployment`
attribute `replication` settings of:

- `inherited` (default)

- `onShutdown`

- `onRequestEnd`

- `none`

The `replication` attribute for stateful session beans cannot be configured in
Application Server Control. The `inherited` value is never displayed and the value
cannot be reset to `none`.

To work around this problem, for all stateful session beans, you must manually
configure the `orion-ejb-jar.xml` file `session-deployment` element
`replication` attribute.

5.3.5 EJB 3.0 Entities and Application Server Control

When you deploy EJB 3.0 entities to OC4J, you cannot manage them using Application
Server Control: when you use Application Server Control to view your EJB module,
the Entity Beans area will display "No entity beans found".

You can manage all other EJB 3.0 beans such as session beans. For example, if you
deploy an EJB module that contains both EJB 3.0 entities and EJB 3.0 session beans,
your session beans will be visible through Application Server Control.

5.3.6 Entity and Session Deployment Attribute tx-retry-wait

The `orion-ejb-jar.xml` file `entity-deployment` and `session-deployment`
element tx-retry-wait attribute is not in `orion-ejb-jar-10_0.xsd` (nor in
`orion-ejb-jar.dtd`).

You can still use this attribute in your `orion-ejb-jar.xml` file but if you do, do not
configure OC4J to perform XML file validation (using the `-validateXML` option on
the OC4J startup command line).

5.4 Web Services

This section describes release notes for Web Services. It covers the following topics:

5.4.1 Long File Names Cause Deployment to Fail

If the combined length of the generated file and directory names passes a certain size limit, then deployment will fail and throw an error. This size limit varies for different operating systems.

> **Note:** You can avoid this problem by upgrading to a more recent version of the J2SE 5.0 JDK (`jdk-1_5_0_06` or later).

The length of the names is controlled by WebServicesAssembler and the deployment code. WebServicesAssembler generates file names based on the method name in the

Java class or the operation name in the WSDL. The deployment code creates directories for code generation based on the names of the EAR and the WAR files.

To avoid the generation of file and directory names that are too long, limit the number of characters in the following names to a reasonable length.

- Method names in Java classes

- Operation names in the WSDL

- Directory name for the location of the OC4J installation (also be aware that the JDeveloper's built-in OC4J instance is typically placed in a directory below the JDeveloper installation)

- File name for a WAR file

- File name for an EAR file

5.4.2 SoapFaultException Will Not Invoke a Handler's handleFault Method

On the server, a `SoapFaultException()` thrown by an implementation class will not invoke a handler's `handleFault()` method. The `handleResponse()` method is called instead.

5.4.3 Clients Cannot Deserialize SOAP-Encoded anyType Arrays

Clients of SOAP-encoded services are not able to deserialize arrays of type `anyType`.

5.4.4 Arrays in Document-Literal Encoding May Not be Supported when Mapped to a Single Array Parameter

Arrays may not be supported in document-literal encoding when mapped directly to a Java method parameter. This issue has been seen in DII and WSIF clients.

It also occurs in document-literal Web services that map `base64Binary` (or `hexBinary`) arrays to the type `byte[][]`.

There are two possible ways to work around this issue:

- Keep the wrapper by specifying the WebServicesAssembler argument `unwrapParam="false"`.

- Use RCP-encoded or RPC-literal styles.

5.4.5 NLS Characters in SYS.XMLTYPE Values May Not be Supported

In Database Web Services, NLS characters that occur in a `SQL SYS.XMLTYPE` value may not be properly handled.

5.4.6 Self Referential WSDL Imports Fail to Load in the Test Page

The test page (Web Services Home Page) fails to load when using self-referential WSDL imports to the same application. For example:

```
location="http://samebox:8888/sameapp/import.wsdl"
```

Since the WSDL is available locally in the application, it should be referenced with a relative path instead. For example:

```
location="./import.wsdl"
```

5.4.7 SOAP 1.2 Results May Not be Properly Deserialized

In certain cases, the SOAP 1.2 response may not be properly deserialized, resulting in an element-name-mismatch exception. Specifically, this happens if the Web services returns output parameters and a result value, but this result element does not immediately follow after an `http://www.w3.org/2003/05/soap-rpc` result element.

5.4.8 WSIF Mapping of Nillable XSD Types

WSIF invocations will map the primitive and nillable XML schema types to primitive Java types. This does not permit the representation of XML nil values.

As a work around, you may want to use SOAP-encoded XML types in the WSDL.

5.4.9 Support for NLS Characters in the WSDL

NLS characters that occur in names in the WSDL, such as in the name of a service, port type, operation, binding or port, are not supported. This may also result in errors on the test page (Web Services Home Page).

5.4.10 Multiple Service Elements in Top Down Web Service Assembly

WebServicesAssembler does not support multiple service elements for the `topDownAssemble` command.

5.4.11 Multiple Message Formats in a WSDL Application

Multiple message formats, such as RPC-encoded and document-literal, are not supported in a single Web application.

5.4.12 Invalid Configuration Not Detected for EJB 2.1 Web Services

EJB 2.1 Web services will be deployed during server side code generation even if the configuration is incorrect.

5.4.13 Schema Features Limitations

This section describes Web Services schema features limitations. It covers the following topic(s):

- Section 5.4.13.1, "Schema Features that are Mapped to a SOAPElement"

- Section 5.4.13.2, "Derived complexTypes Are Not Handled Properly"

- Section 5.4.13.3, "RPC Encoded Does Not Support Complex Types With Attributes"

- Section 5.4.13.4, "XML Types xsd:choice and xsd:group are Not Supported for Proxy or Top Down Web Service Assembly"

5.4.13.1 Schema Features that are Mapped to a SOAPElement

If any of the following schema features are encountered in the WSDL, they will be mapped to a `SOAPElement`.

- Any model group with multiple `xsd:any` elements

- `xsd:choice` elements

- Mixed content

- Substitution groups

- A type with multiple `xsd:anyAttribute`

5.4.13.2 Derived complexTypes Are Not Handled Properly

If a `complexType` derives from another by adding some attributes, then once the complexType is run through the OC4J WSDL2Java tool, all of the attributes in the subtype will be deleted. If the subtype does not have additional elements, it will be presented as a `SOAPElement` in the generated Java code.

If you are able to edit the WSDL, then you can work around this problem in either of the following ways:

- Move the attribute definitions from the sub type to the supertype.

- Avoid using type extensions.

5.4.13.3 RPC Encoded Does Not Support Complex Types With Attributes

If the schema contains a binding with an RPC-encoded message format and WebServicesAssembler encounters a complexType with attributes, then it will throw an "unsupported type encountered" error message.

5.4.13.4 XML Types xsd:choice and xsd:group are Not Supported for Proxy or Top Down Web Service Assembly

If you are assembling Web Services top down or assembling Web service proxies, WebServicesAssembler cannot consume WSDLs that contain the `xsd:choice` or `xsd:group` XML types. If you want to consume a WSDL that contains these XML types, set the WebServicesAssembler `dataBinding` argument to `false` and code the `SOAPElement` so that the payload conforms to the schema definition in the WSDL file.

5.4.14 Limitations on Top Down Processing of Type Mappings

You can specify the WebServicesAssembler `ddFileName` argument to define type mappings. A type mapping maps a schema type to an existing Java class and allows an optional custom serializer. In the top down use case, if you do not supply a custom serializer, then WebServicesAssembler will always generate a bean for the type.

The work around for this limitation is to ensure that the existing Java class is in the classpath given to WebServicesAssembler and that the `overwriteBeans` argument is set to `false`.

5.4.15 REST-Enabled Web Services Cannot be Deployed with Application Server Control

Application Server Control cannot successfully deploy EAR files containing REST-enabled Web services. Instead of using Application Server Control, you can use JDeveloper, Ant, or `admin_client.jar` to deploy the EAR file.

5.4.16 Explicit HTTP Data Chunking is Not Supported

Enabling chunked data transfer for HTTP as described in Chapter 13 of the *Oracle Application Server Web Services Developer's Guide* by explicitly setting `DO_NOT_CHUNK` or `CHUNK_SIZE` properties will not have any effect. However, chunking will still be implicitly enabled when using attachments.

5.4.17 Runtime Exception Masked By java.io.NotSerializableException

When the Web Service client is invoked by an EJB, the RMI protocol requires that the client parameter, return, and exception implement `java.io.Serializable`.

In the current release, however, the `oracle.j2ee.ws.common.util.localization.LocalizableSupport` class does not implement `java.io.Serializable`. Consequently, exceptions thrown by a Web service client invoked by EJB are not properly returned to the invoker. Instead, the invoker receives the description below.

```
Error deserializing return-value: writing aborted;
java.io.NotSerializableException:
/@ oracle.j2ee.ws.common.util.localization.LocalizableSupport; nested /
exception is:
java.io.WriteAbortedException: writing aborted;
java.io.NotSerializableException:
/@ oracle.j2ee.ws.common.util.localization.LocalizableSupport/
java.rmi.UnmarshalException: Error deserializing return-value: writing
aborted; java.io.NotSerializableException:
```

5.4.18 Get NodeLists by Using getFirstChild and getNextSibling Instead of getChildNode

You may see a performance degradation when iterating over a `NodeList` obtained by using `node.getChildNode`. This degradation will only be significant for `NodeLists` with very long lengths.

Instead of using the `NodeList` obtained by `node.getChildNodes`, the current Oracle XDK implementation offers an optimization of navigating a list of child nodes by using `node.getFirstChild` and looping over `node.getNextSibling`. The following code sample illustrates this technique.

```
Node n = ...;
if (n.hasChildNodes()) {
   for(Node nd=n.getFirstChild(); nd!=null; nd=nd.getNextSibling()){
        nd.getValue(); // do something with nd
      }
}
```

5.4.19 Performing Client Call-out to Web Services from the Database

The Web services call-out utility allows a client to access Web services from the database. Web services call-out can be employed by PL/SQL clients, SQL statements and Java in the database. This utility is based on JPublisher 10g Release 2 (10.2) and Oracle Web Services 10g Release 3 (10.1.3).

Two versions of the utility are available, based on whether your target database is the Oracle 10g database (Release 10.1 or 10.2), or a pre-Oracle 10g database.

The call-out utility is available at the following Web site under the heading "*Database as Web Services consumer: Calling-out external Web services*".

http://www.oracle.com/technology/sample_
code/tech/java/jsp/dbwebservices.html

The following sections describe how a client in the database can perform Web services call-out:

- Section 5.4.19.1, "How to Perform Web Services Call-Out using Static Proxy and JPublisher"

- Section 5.4.19.2, "How to Perform Web Services Callout using Dynamic Invocation Interface and the SYS.UTL_DBWS Utility"

5.4.19.1 How to Perform Web Services Call-Out using Static Proxy and JPublisher

The JPublisher 10g command line option -proxywsdl can be used to generate database-side Java and PL/SQL wrappers from the WSDL file of a Web service. To allow JPublisher to generate and load wrappers for Web service clients into the database, the dbwsa.jar and dbwsclient.jar files must be present in the classpath and inside the database respectively.

The following procedure sets up the environment and the database for JPublisher-supported Web service call-out. This procedure needs to be performed only once.

1. Download and install the JPublisher 10g Release 2 (10.2) if it is not already on your system.

    You can obtain the JPublisher release from the following Web site:

    http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

2. Add the dbwsa.jar to the directory or ORACLE_HOME/sqlj/lib.

3. Set up the appropriate JDK as the Java VM and Java compiler.

    The version of the JDK must be the same as the Java VM in the target database:

    - Use the JDK 1.4 for the Oracle 10g (Release 10.1 or 10.2) database.

    - Use the JDK 1.3 for the Oracle 9.2 database.

4. Add dbwsa.jar file to the classpath environment variable.

5. Load the dbwsclient.jar file either into the SYS schema or into the schema where the Web service client will be invoked.

    For example, the following loadjava command will load the dbwsclient.jar file into the SYS schema.

    ```
    %loadjava -u sys/change_on_install -r -v -f -s -grant public -noverify
    -genmissing dbwsclient.jar
    ```

    The following loadjava command illustrates how to load the dbwsclient.jar file into a specific schema.

    ```
    % loadjava -u scott/tiger -r -v -f -noverify -genmissing dbwsclient.jar
    ```

Example

The following example illustrates how to generate Java and PL/SQL wrappers for a Web service client and then invoke it by using SQL statements. The example follows these general steps:

1. Identify the Web service you want to invoke.

2. Call JPublisher with the appropriate options to generate the client proxy, the PL/SQL and Java wrappers, and load them into the database.

    A JPublisher command to do this would include the required -proxywsdl and -user options. The command could also include the optional -endpoint, -httpproxy, -sysuser, -dir, and -proxyopts options. For example:

```
% jpub -user=username/password -sysuser=superuser_name/superuser_password
-proxywsdl=WSDL_URL -endpoint=Web_services_endpoint
```

3. Invoke the Web service by using the appropriate SQL, PL/SQL, or Java commands.

The following command lines deploy the Web service `javacallout` to an OC4J instance. The `samples/javacallout.wsdl` file is the WSDL file for the Web services.

```
% java -jar $J2EE_HOME/admin.jar ormi://localhost oc4jadmin welcome -deploy -file
samples/javacallout.ear -deploymentName javacallout

% java -jar $J2EE_HOME/admin.jar ormi://localhost oc4jadmin welcome -bindWebApp
javacallout javacallout http-web-site /javacallout
```

The following command creates the Web service client and its Java and PL/SQL wrappers in the subdirectory `tmp`, then loads the wrappers into the database.

```
% jpub -user scott/tiger -sysuser sys/change_on_install
-proxywsdl=sample/javacallout.wsdl
-endpoint=http://localhost:8888/javacallout/javacallout -dir=tmp
```

This command produces the following output:

```
tmp/HelloServiceEJBJPub.java
tmp/plsql_wrapper.sql
tmp/plsql_dropper.sql
tmp/plsql_grant.sql
tmp/plsql_revoke.sql
Executing tmp/plsql_dropper.sql
Executing tmp/plsql_wrapper.sql
Executing tmp/plsql_grant.sql
Loading tmp/plsql_proxy.jar
```

You can invoke the PL/SQL functions provided in `tmp/plsql_wrapper.sql`. Each PL/SQL function corresponds to an operation in the Web service. For example, if your Web service is available at the following endpoint:

```
http://localhost:8888/javacallout/javacallout
```

then you can issue the following SQL command.

```
SQL> select jpub_plsql_wrapper.sayhello('hello') from dual;
```

The command will return the following output.

```
JPUB_PLSQL_WRAPPER.SAYHELLO('HELLO')
----------------------------------
HELLO!! You just said :hello
```

For more information about JPublisher call-out support, see "Web service Call-out using Dynamic Invocation Interface and the SYS.UTL_DBWS Utility" in *Oracle Database JPublisher User's Guide*. This is available from the following Web site.

```
http://www.oracle.com/technology/software/tech/java/sqlj_
jdbc/index.html
```

5.4.19.2 How to Perform Web Services Callout using Dynamic Invocation Interface and the SYS.UTL_DBWS Utility

A PL/SQL user can invoke Web services using the PL/SQL package `SYS.UTL_DBWS`. This package provides wrapper functions for the JAX-RPC Dynamic Invocation Interface (DII) APIs.

The following procedure applies to Oracle Database 10g Release 1 or later. The procedure sets up the database for Web service callout using `SYS.UTL_DBWS`, and refreshes the utility with more recent features provided in this download.

1. Load the `dbwsclient.jar` file into the database, as described in "How to Perform Web Services Call-Out using Static Proxy and JPublisher".

2. Execute `utl_dbws_decl.sql` and `utl_dbws_body.sql` SQL scripts as `SYS`.

   You can now use `SYS.UTL_DBWS` to call Web services.

Example

The call-out utilities download includes the SQL scripts `samples/test-plsql-dii.sql` and `samples/test-plsql-dii2.sql`.

Both scripts invoke the `sayHello` operation defined in `javacallout.wsdl`. Examine the two scripts for their correspondences with the WSDL file. Both scripts produce the following output; the phrase `PL/SQL DII client return` is produced by the client code.

```
PL/SQL DII client return HELLO!! You just said :hello
```

For more information on the `SYS.UTL_DBWS` utility, see *Oracle Database PL/SQL Packages and Types Reference* available from the following Web site.

http://www.oracle.com/technology/documentation/database10g.html

5.4.20 genWsdl Command Does not Preserve Order of Variables

If you use the WebServicesAssembler `genWsdl` command or Ant task, then the variables in the generated WSDL file will be in a different order from those in the original Java class files. This can cause problems if you continue to modify your Java code after generating the WSDL file. For example, clients may no longer be able to access the server's Web services.

To work around this problem:

1. Generate the WSDL file with `genWSDL`.

2. Edit the generated WSDL file to place the variables in the desired order.

3. If you are assembling the Web service bottom up, then replace the original WSDL file in the EAR/WAR file with the edited WSDL file.

5.4.21 genValueType Command Does Not Validate Restrictions

In JAX-RPC Web services, the `genValueType` command generates bean classes for data types but does not validate restrictions in the XSD file.

For example, the following XSD fragment defines a `SerialNumber` data type based on string that restricts its length to 16 characters. When `genValueType` creates the bean classes, it does not validate the length.

    . . .

```
         xsd:element name="SerialNumber">
              <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                  <xsd:length value="16"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:element>/
   ...
```

This is a known limitation: the Oracle Web Services JAX-RPC stack does not enforce most restrictions. To work around this problem, write your own bean classes instead of using generated bean classes.

5.4.22 XML Serialization Does Not Accept the Array Java Data Type for Document-Literal-Bare Message Format

If you attempt to expose a Java method that uses the `array` Java type as a Web service that uses the `document-literal-bare` (unwrapped) message format, then the serialization of the method will fail.

For example, assume that you want to expose the method

```
public SearchResult[] WebServiceClass()
```

which has the following definition:

```
   public class SearchResult implements Serializable {
   public SearchResult() { }
   private String name;
   private String value;
   public void setName(String name) { this.name = name; }
   public String getName() { return name; }
   public void setValue(String value) { this.value = value; }
   public String getValue() { return value; }
    }

    public SearchResult[] WebServiceClass(String testTriger) {
            System.out.println(testTriger);
            return STATIC_RESULT_A;
        }
```

If you create the Web service bottom up, the WSDL file will contain the following XML for the `SearchResult` method:

```
      ...
     <sequence>
      <element name="result" type="tns:SearchResult" nillable="true"
minOccurs="0"
      maxOccurs="unbounded"/>
      </sequence>
      ...
```

However, on executing the service, the following error will be returned:

```
ERROR OWS-04046 serialization error: java.lang.ClassCastException:
```

The work around for this issue is to use `doc-literal-wrapped` message format.

5.5 Web Services Security

This section describes release notes for Web Services Security. It covers the following topic(s):

5.5.1 Stale Indirect User Accounts Must be Removed Manually

In release 10.1.3, you must use Application Server Control to obfuscate the keystore, signature key, and encryption key passwords. During obfuscation, an indirect user account is created in the `system-jazn-data.xml` file.

If you undeploy the application, these indirect user accounts are not removed. You must manually delete the them by using Application Server Control.

The following list describes how you can identify the names of indirect user accounts for global-level and port-level keystores and keys.

- For a port-level keystore, the name of the indirect user account is created with the following format:

  applicationName.*portName*.`keystore`.*actual-keystore-name*

  For example:

  `my-security-sample.myport.keystore.myks.jks`

- For a global-level keystore, the name of the indirect user account is created with the following format:

  `default.keystore.`*actual-keystore-name*

  For example:

  `default.keystore.myks.jks`

- For port-level keys, the name of the indirect user account is created with the following format:

  applicationName.*portName*.`key`.*actual-key-alias*

  For example:

  `my-security-sample.myport.key.mysignkey`

- For global-level keys, the name of the indirect user account is created with the following format:

  `default.key.`*actual-key-alias*

  For example:

  `default.key.mysignkey`

5.6 OC4J Services

This section describes release notes for OC4J Services. OC4J Services include: Java Naming and Directory Interface (JNDI), Oracle Enterprise Messaging Service (OEMS), Data Sources, Remote Method Invocation (ORMI and IIOP), OC4J Transaction Support, Java Object Cache (JOC), and XML Query Service (XQS).

The section contains release notes for the following OC4J Services:

5.6.1 JNDI

This section describes release notes for JNDI. It covers the following topic(s):

5.6.1.1 New Package Names for RMI and Application Client Initial Context Factories

In this release, note the following new package names for the initial context factories:

- `oracle.j2ee.rmi.RMIInitialContextFactory`
- `oracle.j2ee.naming.ApplicationClientInitialContextFactory`

5.6.1.2 These Environment Properties Are No Longer Supported

The following environment properties are no longer supported as of release 10.1.3:

`dedicated.connection`

`dedicated.rmicontext`

In release 10.1.3, the known ORMI /JNDI bugs that required these flags have been resolved. To enable client-side ORMI load-balancing in 10.1.3, use the `oracle.j2ee.rmi.loadBalance` property described in the "Load Balancing" section of the JNDI chapter of the *Oracle Containers for J2EE Services Guide*.

5.6.1.3 Context Factory Restructuring

The package structure for context factories provided by previous releases of OC4J is deprecated, and is replaced by a more consistent naming structure. The following context factories are deprecated in release 10.1.3:

- `com.evermind.server.rmi.RMIInitialContextFactory`
- `com.evermind.server.ApplicationClientInitialContext Factory`
- `com.oracle.iiop.server.IIOPInitialContextFactory`

For the new context factory names that replace the deprecated ones, see the `java.naming.factory.initial` initial context property described in the "Constructing a JNDI Context" section of the JNDI chapter of the *Oracle Containers for J2EE Services Guide*.

5.6.1.4 Objects that Implement javax.naming.Referenceable Interface

OC4J JNDI in 10.1.3 now provides full support for binding objects that implement the `javax.naming.Referenceable` interface.

5.6.1.5 Local Host Not Supported

The `java.naming.provider.url` JNDI property does not support the value `localhost` when a remote client connects to an application server instance that is managed by OPMN. The value must be the complete hostname or IP Address. This does not affect clients that connect to standalone application server instances.

5.6.2 Oracle Enterprise Messaging Service (OEMS)

This section describes release notes for the Oracle Enterprise Messaging Service (OEMS). It covers the following topic(s):

- Section 5.6.2.1, "Special Considerations For Undeploying the Default Instance of the Oracle gJRA Resource Adapter"

- Section 5.6.2.2, "OC4J May Fail to Restart after Abnormal OC4J Shutdown"

- Section 5.6.2.3, "getconfigProperties() Lists Some Unsupported Properties"

- Section 5.6.2.4, "XA-styled JMS Connections Not Supported Between OC4J Versions"

- Section 5.6.2.5, "Poor Performance When Persisting Messages to a Database"

5.6.2.1 Special Considerations For Undeploying the Default Instance of the Oracle gJRA Resource Adapter

OC4J cannot be started with OracleASjms, the pre-packaged standalone JMS Connector, undeployed without certain changes. This note deals with additional changes necessary to start OC4J while the default instance of the Oracle gJRA resource adapter, OracleASjms, is undeployed. For general undeployment of a resource adapter, see the Oracle Containers for J2EE Services Guide. The following additional changes must be made:

- In `$J2EE_HOME/config/application.xml`, comment out the following lines:

  ```
  <web-module id="jmsrouter_web" path="../../home/applications/jmsrouter.war" />
  <ejb-module id="jmsrouter_ejb" path="../../home/applications/jmsrouter-ejb.jar" />
  ```

- In `$J2EE_HOME/config/default-web-site.xml`, comment out the following line

  ```
    <web-app application="default" name="jmsrouter_web" root="/jmsrouter"
  load-on-startup="true" />
  ```

If these changes are made, OC4J may be started, but the OracleAS JMS Router will not work.

To reinstate the JMS Router:

1. Fully redeploy the OracleASjms resource adapter instance.

2. Uncomment the lines mentioned above in
   `$J2EE_HOME/config/application.xml`
   and
   `$J2EE_HOME/config/default-web-site.xml`.

When OC4J is restarted, the OracleAS JMS Router should be available.

5.6.2.2 OC4J May Fail to Restart after Abnormal OC4J Shutdown

If you encounter OC4J JMS Server startup problems after an abnormalOC4J shutdown, first check that no other OC4J JMS Server is running and using the same persistence files. Then remove any `.lock` files from the

```
ORACLE_HOME/j2ee/instance_name/persistence
```

directory and try restarting again.

If problems persist, confirm that the `jms.xml` file is valid.

If problems still persist, remove the `jms.state` file from the persistence directory and try again, but be aware that removing this file may result in loss of transaction information. For additional information, see the section "Abnormal Termination" in the "Oracle Enterprise Messaging Service" chapter of the *Oracle Containers for J2EE Services Guide*.

5.6.2.3 getconfigProperties() Lists Some Unsupported Properties

The list of properties returned by the JMS Administrator MBean's `getconfigProperties()` method includes the following properties that are neither documented nor supported:

- `oc4j.jms.checkPermissions`

- `oc4j.jms.j2ee14`

- `oc4j.jms.noJmx`

- `oc4j.jms.printStackTrace`

- `oc4j.jms.rememberAllXids`

5.6.2.4 XA-styled JMS Connections Not Supported Between OC4J Versions

Oracle Application Server does not support XA-styled JMS connections between different versions of OC4J.

5.6.2.5 Poor Performance When Persisting Messages to a Database

The OEMS database persistence feature performs poorly if the queue tables are configured to use old compatibility modes. The compatibility mode is set using the `compatible` parameter. For example:

```
DBMS_AQADM.CREATE_QUEUE_TABLE(
        Queue_table            => 'demoTestQTab',
        Queue_payload_type     => 'SYS.AQ$_JMS_MESSAGE',
        sort_list              => 'PRIORITY,ENQ_TIME',
        multiple_consumers     => false,
        compatible             => '8.1.5');
```

The `compatible` parameter should only be set to older modes if there are issues migrating to the new schema layouts or if existing queue tables cannot be exported to the new schema layouts.

When using the latest schema layouts, set the `compatible` parameter to `10.0.0.0.0` or omit the parameter to use the default compatibility mode. Either option ensures that the most efficient schema and locking mechanisms are used.

5.6.3 Data Sources

This section describes release notes for Data Sources. It covers the following topics:

5.6.3.1 New Syntax for Data Source Configuration

The data sources subsystem has been completely rewritten. Part of the rewrite includes a new syntax for the configuration using the `data-sources.xml` file. The pre-10.1.3 syntax is still supported but users are encouraged to use the new syntax. Also users are encouraged to convert their existing application `data-sources.xml` files to the new syntax. See the *Oracle Containers for J2EE Services Guide* for details on converting `data-sources.xml`.

5.6.3.2 OracleConnectionCacheImpl Deprecated

The class `oracle.jdbc.pool.OracleConnectionCacheImpl` has been deprecated because it does not support multiple schemas. When defining the `factory-class` for connection factories and `data-source-class` for native data sources, use `oracle.jdbc.pool.OracleDataSource` instead of `oracle.jdbc.pool.OracleConnectionCacheImpl`.

5.6.3.3 Converting Existing Data Sources to Release 3 Format

The OC4J 10.1.3 implementation understands the 10.1.3 and the pre-10.1.3 (10.1.2 and 9.0.4) formats of the `data-sources.xml` file. For an application that was previously used in a pre-10.1.3 OC4J implementation and contains its own `data-sources.xml` file, the OC4J 10.1.3 implementation automatically converts the `data-sources.xml` file from the pre-10.1.3 format to the 10.1.3 format when you use the Application Server Control Console to change anything in the `data-sources.xml` file, such as modifying an existing data source or creating or deleting a data source.

With an active OC4J instance in a standalone environment, you can alternatively use `admin.jar` with the following syntax to manually convert a pre-10.1.3 `data-sources.xml` file to the 10.1.3 format. This is discussed in the *Oracle Containers for J2EE Configuration and Administration Guide*. The guide does not mention that you can specify an ORMI URL only when OC4J is running or that the ORMI URL is optional.

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-convertDataSourceConfiguration old-data-sources.xml new-data-sources.xml
```

You can also convert a `data-sources.xml` file before deployment, without a running OC4J instance. The syntax for this offline conversion is as follows:

```
java -jar admin.jar -convertDataSourceConfiguration old-data-sources.xml
```

new-data-sources.xml

> **Notes:**
>
> - If you include the ORMI port, then OC4J must be running. When OC4J is not running, you must omit the ORMI URL from the admin.jar command line.
>
> - If you do not include the ORMI port, then the admin.jar command will work either way, that is, with OC4J running or with OC4J not running.
>
> - The admin.jar utility works only in the standalone OC4J environment. This utility is installed in the Oracle Application Server environment, but does not work in an OPMN-managed environment.
>
> - The newer admin_client.jar utility works in both environments, standalone and managed Oracle Application Server. However, the admin_client.jar utility does not convert data-sources.xml files.

Check for Consistency Between Your Application and the New data-sources.xml File

After conversion, whether manual or automatic, visually inspect the new `data-sources.xml` file and confirm that there is consistency between your application and the new file regarding the JNDI location used to refer to a data source. This is advisable because the new file may contain data source definitions that are not used.

This happens because the old format uses multiple location attributes (such as `location`, `ejb-location`, `xa-location`, and so on). The conversion to the new 10.1.3 format creates a separate data source in the new `data-sources.xml` file corresponding to each location attribute specified in the old `data-sources.xml` file. In most cases, client applications will only use the data source defined by either the `location` or `ejb-location` attribute. But we cannot be sure of this. Therefore, the converted `data-sources.xml` may have definitions that are not used by the applications and can be removed from the file.

You can also refer to examples of the new `data-sources.xml` format in the "Data Sources" chapter of the *Oracle Containers for J2EE Services Guide*.

5.6.3.4 Configuring OC4J instance for oci drivers

If any application uses oci drivers for a datasource connection then, add the following *<environment>* section in `opmn.xml` for that particular OC4J instance. For example, if the application is deployed to the default home instance, then configure the OC4J instance as follows:

```
<ias-component id="OC4J">
   <environment>
      <variable id="LIBPATH" value="$ORACLE_HOME/lib32" />
   </environment>
     <process-type id="home" module-id="OC4J" status="enabled">
```

5.6.3.5 SQLServer Named Instance Definition Corrected

A SQL Server named instance defined outside of Java code, i.e., in data-sources.xml in OC4J, should have only one backward slash('\') separating the server name and the instance name.

This is incorrectly documented in the Data Sources chapter of the *Oracle Containers for J2EE Services Guide*. For further information, see the release note at Section 5.9.3.3, "SQLServer Named Instance Definition Corrected" on page 5-52.

5.6.3.6 Use the Current Oracle JDBC Drivers when Using Shared Libraries

In OC4J 10.1.3.0, some Orion CMP features may not work as designed if the current Oracle JDBC Drivers that are shipped with OC4J are not used, and the JDBC Driver libraries are added to an OC4J instance using the Shared Library feature. In such situations, different JDBC library versions will clash.

5.6.3.7 Fast Connection Failover Is Disabled When Using OracleXADataSource

In earlier versions of release 10.1.3, OC4J incorrectly disables Fast Connection Failover (FCF) when the connection factory for a connection pool is configured to use `oracle.jdbc.xa.client.OracleXADataSource`. One-off patch number 5001421 has been created to fix the problem. This patch can be downloaded at the Oracle MetaLink.

5.6.3.8 Upgrading the Oracle THIN JDBC Driver

The JDBC THIN driver cannot be upgraded or changed at the Oracle Application Server-instance level due to component dependencies. JDBC THIN driver upgrades must be completed for each OC4J instance using the Shared Library feature.

Instructions for upgrading the JDBC THIN driver are located on the OTN How To site:

http://www.oracle.com/technology/tech/java/oc4j/1013/how_
to/index.html

5.6.4 OC4J Transaction Support

This section describes release notes for OC4J Transaction Support. It covers the following topics:

- Section 5.6.4.1, "Change the Default JTA Recovery Password Immediately"

- Section 5.6.4.2, "New Configuration File for Transaction Manager"

- Section 5.6.4.3, "The In-DB Coordinator Is Deprecated"

- Section 5.6.4.4, "The Mid-Tier Coordinator Does Not Use a Persistent Store By Default"

- Section 5.6.4.5, "DMS must be enabled to obtain JTA statistics"

- Section 5.6.4.6, "Transaction Propagation Between 10.1.3 Instances Only"

5.6.4.1 Change the Default JTA Recovery Password Immediately

The default JTA recovery password should be changed immediately. OC4J is shipped with a default password, which should be changed after install. The recovery password is configured in the configuration file `jazn-data.xml`, which is in the `$J2EE_HOME/config` directory. To modify the transaction recovery password, change the credentials value for the user `JtaAdmin` in the `jazn-data.xml` file.

```
<user>
```

```
    <name>JtaAdmin</name>
    <display-name>JTA Recovery User</display-name>
    <description>Used to recover propagated OC4J transactions</description>
    <credentials>!newJtapassword</credentials>
</user>
```

Even if OC4J is configured to use a security service other than JAZN, such as OID the transaction recovery password must still be configured in `jazn-data.xml`.

5.6.4.2 New Configuration File for Transaction Manager

All transaction-manager-related configuration is now done in the `transaction-manager.xml` file.

5.6.4.3 The In-DB Coordinator Is Deprecated

The use of the in-database transaction coordinator by OC4J is deprecated as of release 10.1.3. Oracle recommends that the middle-tier transaction coordinator be used going forward.

5.6.4.4 The Mid-Tier Coordinator Does Not Use a Persistent Store By Default

The mid-tier coordinator does not use a persistent store by default. Prior to use in production, the mid-tier coordinator should be configured to use a persistent store which will enable transaction recovery.

5.6.4.5 DMS must be enabled to obtain JTA statistics

To obtain JTA statistics, ensure that DMS is enabled.

5.6.4.6 Transaction Propagation Between 10.1.3 Instances Only

Transaction propagation is only supported between 10.1.3 instances.

5.6.5 RMI

This section describes release notes for OC4J Remote Method Invocation (RMI and IIOP). It covers the following topics:

- Section 5.6.5.1, "RMI Recommendations"
- Section 5.6.5.2, "Excessive ORMI Connections Created"
- Section 5.6.5.3, "Workaround for HTTP Tunnelling Failover"
- Section 5.6.5.4, "Incorrect "Provider URL..." Error Message"

5.6.5.1 RMI Recommendations

In this release, note the following recommendations:

- Environment variables `dedicated.connection` and `dedicated.rmicontext` are not required for lookup using EJBs in 10.1.3

  The `dedicated.connection` environment variable is still required for EJBs hosted in 10.1.2 being looked up from 10.1.3 - Bug 4895256

- The RMI port may not be released immediately sometimes. - Bug 4892487

- Old tunneling is deprecated. Use the new URL, as described in the "Configuring ORMI Tunneling through HTTP" section of the "RMI" chapter of the *Oracle Containers for J2EE Services Guide*

■ Old package names for context factories deprecated and new names are recommended to be used, as described in the "Constructing JNDI Context" section of the "JNDI" chapter of the *Oracle Containers for J2EE Services Guide*

5.6.5.2 Excessive ORMI Connections Created

Setting the JNDI property `oracle.j2ee.rmi.loadBalance` to either `context` or `lookup` currently creates separate ORMI connections for each call to new `InitialContext`. Closing the context does not cause the connection to be closed. Doing this repeatedly will result in performance degredation. - Bug 4902304

5.6.5.3 Workaround for HTTP Tunnelling Failover

The following workaround is necessary for HTTP Tunnelling failover to work in iAS mode. This workaround applies to iAS mode, where the provider URL points to OHS using `mod_oc4j`. The workaround is not needed in standalone mode, where the provider URL lists multiple OC4J instances.- Bug 4599521

The workaround must be done for all OC4J server instances in the cluster, such as home1, home2, and so on.

1. Go to Application Server Control Console.

2. Select a server instance, such as "home".

3. In the instance window, select the Applications tab.

4. Select the "default" application.

5. In the Application:default window, select the Administration tab.

6. Select the Clustering Properties task.

7. Select the "Override parent application clustering settings" radio button. Specify Clustering Enable.

8. Click the OK button.

9. Repeat for each OC4J server instance.

10. Add the `<distributable />` element to the `<ORACLE_HOME>/j2ee/home/default-web-app/WEB-INF/web.xml` file.

11. Restart the server using `opmnctl stopall` and then `opmnctl startall`.

5.6.5.4 Incorrect "Provider URL..." Error Message

In certain cases when there is something wrong with the provider url format, the following incorrect error message is displayed:

```
" Provider URL must be of the form
[opmn:]corbaname::host:port#/appname"
```

The URL format in the error message is incorrect. The correct URL format is:

```
 [opmn:]corbaname::host:port#[instancename#]appname
```

5.6.6 XQS

This section describes release notes for XML Query Service (XQS). It covers the following topic(s):

■ Section 5.6.6.1, "Implementation Restriction on the fn:doc() and fn:collection() Functions"

- Section 5.6.6.2, "Incorrect XQS Type-Checking Example"

- Section 5.6.6.3, "XQS Schema Import Support"

5.6.6.1 Implementation Restriction on the fn:doc() and fn:collection() Functions

The only arguments that the current implementation of the built-in XQuery functions `fn:doc` and `fn:collection` support are URLs with the `"file"` protocol that specify a path to a file on the local file system, as in this function call:

```
fn:doc("C:/MyDocuments/XQS/myView.xq")
```

The protocol part of the URL is always assumed to be `"file"` and can be omitted.

As an alternative to the `fn:doc()` function, you can use an XQS document function to access a document using any URL that Oracle Application Server supports. For example, a `<document-source>` element has the following configuration:

```
<document-source>
       <function-name prefix="ns"> genericFile </function-name>
</document-source>
```

The XQS function `genericFile()` can be used in a query expression, as follows:

```
declare namespace ns = "…";
declare function ns:genericFile() external;
```

5.6.6.2 Incorrect XQS Type-Checking Example

The XQS chapter in the *OC4J Services Guide* contains an error. The example XQuery expression provided in the "Type-Checking for Input Parameters" section is incorrect. The following example is correct:

```
import schema namespace ns1="urn:namespace_1" at "http://mydomain/myschema.xsd";

declare namespace xqs = "http://xmlns.oracle.com/ias/xqs";
declare function xqs:takeNS1Input($param as element(ns1:InputElement)?) external;

let $in := <ns1:InputElement>...</ns1:InputElement>
let $y := xqs:takeNS1Input($in)
return <result>$y//ns1:Content</result>
```

5.6.6.3 XQS Schema Import Support

The XQS chapter in the *OC4J Services Guide* contains an error. The "XQuery Optional Features" section indicates that the XQuery schema import feature is not supported. XQS does support the schema import feature as defined by the W3C.

5.7 J2EE Connector Architecture (J2CA)

This section describes release notes for J2EE Connector Architecture (J2CA). It covers the following topics:

- Section 5.7.1, "J2CA Lifecycle Issues"

- Section 5.7.2, "Cannot Cast a Connection Handle to a Concrete Type"

- Section 5.7.3, "RAR Name Must Be Unique"

- Section 5.7.4, "Set inactivity-timeout-check in oc4j-ra.xml"

- Section 5.7.5, "Revised Release Note - Stop the Resource Adapter Before Redeploying or Undeploying It"

- Section 5.7.6, "Explicit Configuration Is Necessary For Resource Adapter To Support XA Transaction Recovery"

- Section 5.7.7, "ASControl Changes to Work Manager Thread Pool Not Persisted If <work-manager-thread-pool> Not Defined"

5.7.1 J2CA Lifecycle Issues

- Unable to deploy multiple versions of a standalone RAR - Bug 4253861

- A standalone RAR takes precedence over one in an application. When the same fully-qualified class exists in both a standalone RAR and also in a RAR deployed in an EAR, the class will always be loaded from the standalone RAR. - Bug 4415389

- When stopping a resource adapter, OC4J does not always properly stop dependent applications.

5.7.2 Cannot Cast a Connection Handle to a Concrete Type

OC4J wraps all connection handles with connection handle proxies to perform connection association and therefore connection handles can only be cast to interfaces implemented by the connection handle. An attempt to cast a connection handle to a concrete class will cause a `ClassCastException`.

5.7.3 RAR Name Must Be Unique

`NullPointerException` occurs if an attempt is made to deploy an RAR when there is already an RAR deployed with the same name. - Bug 4884317

5.7.4 Set inactivity-timeout-check in oc4j-ra.xml

Set `inactivity-timeout-check` in the `oc4j-ra.xml` file. Changing the `inactivity-timeout-check` property for an RAR connection pool with ASControl does not work properly. This property should be set to the proper value in the `oc4j-ra.xml` file prior to deploying the resource adapter. - Bug 4455421

5.7.5 Revised Release Note - Stop the Resource Adapter Before Redeploying or Undeploying It

When a resource adapter with active endpoints is redeployed or undeployed without stopping it first, OC4J throws a `DeployerException` due to active endpoints. To work around this issue, stop the resource adapter prior to redeploying or undeploying it. - Bug 4740441

Note:

This release note is revised to include undeployment as well as redeployment.

5.7.6 Explicit Configuration Is Necessary For Resource Adapter To Support XA Transaction Recovery

XA transaction recovery can be configured using Application Server Control using the following steps:

1. In the `Connection Factories` tab accessed from the `Resource Adapter Home` page for the appropriate resource adapter, choose the JNDI location of the connection factory that you want to configure.

2. In the `Options` tab of the resulting `Edit Connection Factory` page, you can do any of the following:

   - Add a new user name.

     After specifying the user name, you can specify a password directly or indirectly. For a direct password, choose `Password` and type the password itself.

     For an indirect password, choose `Indirect Password` and type a key (which might just be the user name, for example). OC4J uses the key to do a lookup in the User Manager (specifically, in the jazn-data.xml file).

   - Change an existing user name or password

5.7.7 ASControl Changes to Work Manager Thread Pool Not Persisted If <work-manager-thread-pool> Not Defined

Changes to work manager thread pool properties from ASControl are not persisted to the `server.xml` file if there is no `<work-manager-thread-pool>` element defined. - Bug 4871940

5.8 OracleAS JAAS Provider and Security

This section describes release notes for the OracleAS JAAS Provider in Release 10.1.3.0.0. It covers the following topics:

■ Section 5.8.14, "AJP13 Protocol Vulnerable to Bypass User Authentication"

5.8.1 COREid Status for 10.1.3.0.0

The initial version of the 10.1.3.0.0 OC4J Release Notes pointed out that as of the 10.1.3.0.0 release, you cannot use the COREid Access security provider for J2EE Web applications deployed in 10.1.3 OC4J. This update is to point out that, in fact, you cannot yet use any functionality through the COREid custom login module. In other words, COREid integration with Web and EJB applications will not be supported until a patch is made available. (Because the Oracle Web Services Manager agent covers integration with COREid, there *is* COREid integration for Web services in the 10.1.3.0.0 implementation.) Refer to OracleMetaLink to check the status of the future 10.1.3.0.0 patch set. - Bugs 4887466, 4772443, 4745790

5.8.2 Restart Application After Configuring Through Security Provider MBean

Whenever a configuration change is made using Application Server Control or the OC4J security provider MBean, the application must be restarted. Until the application is restarted, all other operations of the security provider MBean are invalidated and will return the following message: "The security provider has been changed. Operation temporarily invalidated till application or OC4J restart."

5.8.3 Necessary Permission Grants When Using Security Manager

Users running with a `SecurityManager` in an Oracle Application Server environment should be aware that if an OC4J instance name other than `home` is used, adding the following permission grants to `ORACLE_HOME`/j2ee/`instance_name`/config/java2.policy will be necessary for proper operation of OC4J:

```
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/connectors/OracleASjms/Oracl
eASjms/gjra.jar" {
grant codebase
    permission java.security.AllPermission;
};

grant codebase "file:
${oracle.home}/j2ee/${oracle.oc4j.instancename}/connectors/datasources/datasources
/datasources.jar"
{
    permission java.security.AllPermission;
};
```

(Failure to add these does not compromise security but may hinder OC4J operations.) - Bug 4942880

5.8.4 Indirect Users for Password Indirection

If you choose to use indirect passwords in the OC4J 10.1.3.0.0 implementation, an indirect user is created in the `system-jazn-data.xml` file when you use this feature. Be aware that these indirect user accounts are not removed automatically when an application is undeployed; you must use Application Server Control Console to delete any stale indirect user accounts manually.

5.8.5 JAAS Policy Configuration with Custom Realms

When you use custom realms, and JAAS policies are granted to users or roles in the custom realm, you should do the following:

1. In the `<jazn>` element of your application `orion-application.xml` file, specify a `default-realm` setting of "*custom_realm_name*".

2. Do *not* specify a `location` attribute setting in the `<jazn>` element.

3. Set the `jaas.username.simple` property to "`false`" in `jazn.xml`, using a `<property>` subelement of the `<jazn>` element.

These steps allow the custom realm and its users, roles, and policies to be persisted in `system-jazn-data.xml`.

Note that to use JAAS authorization, in particular to grant permissions to users or roles in a custom realm, the custom realm and its users and groups must be defined and persisted in `system-jazn-data.xml`, not in a `jazn-data.xml` file deployed in the application EAR file.

5.8.6 User Manager Delegation for the File-Based Provider

Before HTTP requests can be dispatched to the target servlet, the OracleAS JAAS Provider `JAZNUserManager` coordinates authentication. `JAZNUserManager` supports the OC4J `UserManager` delegation model, but effectively this applies only to the file-based provider. With delegation, if a user or group is not found at the application-level `JAZNUserManager` instance, the request is delegated to the parent user manager.

Specifically, note the following restrictions and additional details:

- If the application and parent application are both configured to use the file-based provider, delegation goes up through the parent hierarchy as far as necessary, until a parent is not configured to use the file-based provider. Delegation is not propagated beyond that point.

- If the application is configured to use the file-based provider, and the parent is configured to use the LDAP-based provider, an external LDAP provider, or a custom login module, there is no delegation support.

- If the application itself is configured to use the LDAP-based provider, an external LDAP provider, or a custom login module, there is no delegation support.

> **Note:** In OC4J, the `system` application is at the root of the hierarchy, but the `default` application is the default parent of any deployed application. Both use `system-jazn-data.xml` as the user repository.

5.8.7 JNDI Context Pool Timeout Property for Oracle Internet Directory

For the LDAP-based provider (Oracle Internet Directory), the OC4J 10.1.3 implementation includes a new property, `JNDI_CTX_POOL_TIMEOUT`, that you can set in order to specify a timeout for the JNDI context pool. This may be useful, for example, when there is a firewall between the middle tier, including OracleAS JAAS Provider, and the Oracle Internet Directory. The timeout on the firewall connection could be coordinated with the timeout of the directory context.

Set this property through a `<property>` subelement of the `<jazn>` element in the `jazn.xml` file, specifying the timeout in milliseconds. The following example specifies a timeout of 5 seconds.

```
<jazn ... >
   <property name="JNDI_CTX_POOL_TIMEOUT" value="5000">
   ...
```

```
                     </jazn>
```

5.8.8 Miscellaneous OracleAS JAAS Provider and Security Release Notes

- Although it is already stated in the 10.1.3.0.0 OC4J Security Guide, make special note of the fact that the OC4J administration account, `admin` in previous releases, has changed to `oc4jadmin`.

- Security context propagation is supported only between OC4J 10.1.3 instances.

5.8.9 Signature Verification Fails for X.509 Token-Based Authentication On Windows Platforms

On Windows platforms, there is a known issue with X.509 token authentication. The signature verification fails for X.509 token-based authentication.

If X.509 token-based authentication is not used, then the digital signature verification of SOAP body or elements works as expected.

For example, digital signature verification of a web service works as expected if web service is secured with combination of username token or SAML token-based authentication.

Refer to OracleMetaLink to check the status of a future patch to remedy this problem.

5.8.10 J2EE SSO Is Deprecated in the OC4J 10.1.3.0 Implementation

The OC4J 10.1.2 phase 2 implementation provided a lightweight single sign-on feature, characterized by the use of a stateful session EJB to maintain SSO state across multiple Web applications. This J2EE SSO was particularly intended for standalone OC4J, because it did not require a full Oracle Application Server environment as OracleAS Single Sign-On does.

In the OC4J 10.1.3.0 implementation, J2EE SSO still exists but is deprecated and undocumented. It is anticipated that in the OC4J 10.1.3.1 implementation, J2EE SSO will be desupported and replaced by an alternative implementation of equivalent functionality.

5.8.11 Basic Authentication Fallback for Digest Authentication Module

By default in the OC4J 10.1.3.0.0 implementation, the digest authentication module also handles basic authentication if the client sends a basic authentication header. To disable this behavior, so that the digest module would never handle basic authentication, set the `digest.auth.basic.fallback` property to `"true"` as in the example below. (Note that the logic of this setting is counterintuitive and will be reversed in the next release.)

```
<jazn provider="XML" location="jazn-data.xml">
  <property name="digest.auth.basic.fallback" value="true" />
  ...
</jazn>
```

5.8.12 Controlling Caching of JNDI Contexts for LDAP Connections

By default in 10.1.3.x implementations, OC4J uses `java.lang.ref.WeakReference` objects to cache JNDI contexts when using connection pooling for connections to the

LDAP-based provider, and uses JVM garbage collection to garbage-collect the JNDI contexts.

We recommend, however, that you *not* rely on this `WeakReference` and JVM garbage collection functionality. You can disable it with the following property setting:

```
<jazn ... >
   <property name="jndi.ctx_pool.weakref.enable" value="false" />
   ...
</jazn>
```

(The default setting is `"true"`.) When this is disabled, OC4J uses the following properties, which you should set appropriately, to control caching in the connection pool:

- `jndi.ctx_pool.timeout`: Timeout value, in milliseconds, for the LDAP JNDI connection pool. The default is 0, for no timeout. The recommended setting is 3600000 milliseconds, which is one hour.

- `jndi.ctx_pool.threshold_size`: Threshold to limit the number of idle connections in the pool. The default, and recommended initial setting, is 100.

(Both properties are ignored if `jndi.ctx_pool.weakref.enable` is set to `"true"`.)

For example:

```
<jazn ... >
   <property name="jndi.ctx_pool.weakref.enable" value="false" />
   <property name="jndi.ctx_pool.threshold_size" value="100" />
   <property name="jndi.ctx_pool.timeout" value="3600000" />
   ...
</jazn>
```

Notes:

- The `jndi.ctx_pool.weakref.enable` and `jndi.ctx_pool.threshold_size` properties are not documented in the 10.1.3.x *Oracle Containers for J2EE Security Guide*.

- The `jndi.ctx_pool.timeout` property is documented in the 10.1.3.x *Oracle Containers for J2EE Security Guide*, but without mention of the fact that this parameter is ignored if `jndi.ctx_pool.weakref.enable` is set to `"true"` (the default).

5.8.13 JAZN Tool No Longer Includes -clustersupport

The DCM component is no longer supported in Oracle Application Server 10.1.3. Consequently, the `-clustersupport` option is no longer available in the `jazn.jar` administrative tool.

Problem

Using the `-clustersupport` option when invoking `jazn.jar` results in the following error:

```
An error has occured in propagating the changes to the cluster.
java.lang.ClassNotFoundException oracle.security.jazn.smi.DcmUtil
```

This problem occurs because Distributed Configuration Management (DCM), used in prior releases of Oracle Application Server to replicate common configuration information across a cluster, is no longer supported with release 10.1.3. Consequently, the `-clustersupport` option is not supported in the JAZN (jazn.jar) administrative tool.

Solution

Do not use the `-clustersupport` option when invoking `jazn.jar`.

5.8.14 AJP13 Protocol Vulnerable to Bypass User Authentication

When OC4J is running a site using the AJP13 protocol, a security vulnerability exists if a remote attacker can directly access the AJP port on the machine running OC4J. The AJP13 protocol defines an AJP parameter `remote_user`, which is used by OHS to implement `mod_osso`. An attacker can use this parameter to bypass authentication on OC4J. If a user constructs an AJP packet that inserts a valid `remote_user` value as an AJP parameter, the user will be able to access resources that the specified user (remote user) has permission to access.

You must ensure that the system running OC4J does not expose the AJP port to the outside world.

You can protect against the vulnerability in either of the following ways:

- Enable SSL between OC4J and Oracle HTTP Server (preferred). For 10.1.3.x releases, this is documented in the *Oracle Containers for J2EE Security Guide*. For release 10.1.2 or 9.0.4, this is documented in the *Oracle Containers for J2EE Servlet Developer's Guide*.

- Use the `<access-mask>` element (a subelement of `<orion-web-app>`) in `global-web-application.xml` or `orion-web.xml` to restrict access to appropriate host names, domains, or IP addresses. This element is documented in the *Oracle Containers for J2EE Servlet Developer's Guide*.

5.9 Documentation Errata

This section describes known errors in the OC4J documentation in Oracle Application Server 10g Release 3 (10.1.3). It covers the following book(s):

- Section 5.9.1, "Web Services"

- Section 5.9.2, "Oracle Application Server Advanced Web Services Developer's Guide"

- Section 5.9.3, "Oracle Containers for J2EE Services Guide"

- Section 5.9.4, "Oracle Containers for J2EE Developer's Guide"

- Section 5.9.5, "Oracle Containers for J2EE Security Guide"

- Section 5.9.6, "Oracle Containers for J2EE Administration and Configuration Guide"

- Section 5.9.7, "Oracle Containers for J2EE Deployment Guide"

5.9.1 Web Services

This section describes Web Services documentation errata. It covers the following topic(s):

- Section 5.9.1.1, "WebServicesAssembler Command genInterface Does Not Use the use and style Arguments"

- Section 5.9.1.2, "Error in Ant Task for Assembling JMS Web Services"

- Section 5.9.1.3, "Additions to List of Supported Platforms"

5.9.1.1 WebServicesAssembler Command genInterface Does Not Use the use and style Arguments

Book: *Oracle Application Server Web Services Developer's Guide*

Chapter 4, "OracleAS Web Services message Formats", Section: "Selecting Message Formats"

The list of WebServicesAssembler commands that can use the `use` and `style` arguments includes `genInterface`. This is an error. The `genInterface` command cannot use these arguments.

5.9.1.2 Error in Ant Task for Assembling JMS Web Services

Book: *Oracle Application Server Web Services Advanced Developer's Guide*

Chapter 8, "Using JMS as a Web Service Transport", Section: "Assembling a Web Service Bottom Up that Uses JMS Transport"

In Step 1, there is a missing closing angle bracket in the `<oracle:porttype` clause, at the end of the `sendConnectionFactoryLocation` attribute. The `<oracle:porttype` clause should read as follows:

```
...
   <oracle:porttype
       interfaceName="oracle.j2ee.ws.jmstransport.Echo"
       className="oracle.j2ee.ws.jmstransport.EchoImpl"
       >
       <oracle:port
          uri="/echo"
          sendQueueLocation="jms/senderQueue"
          name="EchoPort"
          sendConnectionFactoryLocation="jms/senderQueueConnectionFactory">
       </oracle:port>
...
```

5.9.1.3 Additions to List of Supported Platforms

Book: *Oracle Application Server Web Services Advanced Developer's Guide*

Chapter 3 lists the platforms on which Oracle Web Services is supported. The HP-UX and AIX platforms should be added to this list.

The Oracle Web Services stack is supported on the same platforms as Oracle Application Server and OC4J. For an up-to-date list of the supported platforms for each release, see the following Web site: http://metalink.oracle.com

5.9.2 Oracle Application Server Advanced Web Services Developer's Guide

This section describes errors in the *Oracle Application Server Advanced Web Services Developer's Guide*. It covers the following item(s):

- Section 5.9.2.1, "Auditing and Logging File Path Corrections"

- Section 5.9.2.2, "Incorrect XML in Examples in the "Using Web Service Providers" Chapter"

5.9.2.1 Auditing and Logging File Path Corrections

The *Oracle Application Server Advanced Web Services Developer's Guide* Chapter 6, "Auditing and Logging Messages" lists the paths to the auditing and logging `log.xml` files as:

`<ORACLE_HOME>\log\wsmgmt\logging\log.xml`
and
`<ORACLE_HOME>\log\wsmgmt\audit\log.xml`

These paths are incorrect. The correct paths are:

```
<ORACLE_HOME>\j2ee\<OC4J_instance_
name>\log\wsmgmt\auditing\log.xml
```
and
```
<ORACLE_HOME>\j2ee\<OC4J_instance_
name>\log\wsmgmt\logging\log.xml
```

5.9.2.2 Incorrect XML in Examples in the "Using Web Service Providers" Chapter

The chapter titled "Using Web Service Providers" in the *Oracle Application Server Advanced Web Services Developer's Guide* contains examples with incorrect XML.

Example 10-5

"oracle-webservices.xml Fragment, with a <provider-description> Clause" is not well-formed xml.

- The spaces after slashes in closing tags must be removed:

  Change: `</ wsdl-service-name>` to: `</wsdl-service-name>`

  Change: `</ implementation-class>` to: `</implementation-class>`

- Closing tag is missing a slash:

  Change: `<auditing>...<auditing>` to: `<auditing>...</auditing>`

Example 10-6

"Provider Elements in the web.xml Deployment Descriptor" is well-formed but invalid XML.

The description and display-name elements are out-of-order and mis-positioned according to the JDeveloper 10.1.3.3.0.4157 schema validation:

Change:

```
<servlet-name>LoggerProviderPort</servlet-name>
<display-name>LoggerProviderPort</display-name>
<description>JAX-RPC endpoint  Provider Port</description>
```

to:

```
<description>JAX-RPC endpoint  Provider Port</description>
<display-name>LoggerProviderPort</display-name>
<servlet-name>LoggerProviderPort</servlet-name>
```

5.9.3 Oracle Containers for J2EE Services Guide

This section describes errors in the *Oracle Containers for J2EE Services Guide*. It covers the following item(s):

5.9.3.1 Incorrect URL in Native Data Source Example for Fast Connection Failover

In the *Oracle Containers for J2EE Services Guide* Chapter 4, "Data Sources", the "Enabling Fast Connection Failover in the data-sources.xml File" section, in the native data source example for fast connection failover labelled "Enabling Fast Connection Failover in the data-sources.xml File" is incorrect.

- The url is not correct for a RAC environment.

- The brackets in the <native-data-source> element do not match correctly.

The **INCORRECT** example is as follows:

```
<native-data-source>
   name="nativeDataSource"
   jndi-name="jdbc/nativeDS"
   description="Native DataSource"
   data-source-class="oracle.jdbc.pool.OracleDataSource"
   user="scott"
   password="tiger"
   url="jdbc:oracle:thin:@localhost:1521:oracle">
  <property name="connectionCacheName" value="ICC1"/>
  <property name="connectionCachingEnabled" value="true"/>
  <property name="fastConnectionFailoverEnabled" value="false"/>
</native-data-source>
```

A **CORRECT** example is as follows:

```
<native-data-source
   name="nativeDataSource"
   jndi-name="jdbc/nativeDS"
   description="Native DataSource"
   data-source-class="oracle.jdbc.pool.OracleDataSource"
   user="scott"
   password="tiger"

url="jdbc:oracle:thin:@(DESCRIPTION=
   (LOAD_BALANCE=on)
   (ADDRESS=(PROTOCOL=TCP)(HOST=host1) (PORT=1521))
   (ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))
```

```
         (CONNECT_DATA=(SERVICE_NAME=service_name)))">

  <property name="connectionCacheName" value="ICC1"/>
  <property name="connectionCachingEnabled" value="true"/>
  <property name="fastConnectionFailoverEnabled" value="false"/>
</native-data-source>
```

5.9.3.2 Correction to the OEMS JMS Database Certification Matrix Table

In Table 3-8 OEMS JMS Database Certification Matrix, the last row of the table is not correct, it indicates that OEMS JMS Database 10.1.3 only certifies with DataBase v9.0.1.4, but does not certify with 10.x. In fact, OEMS JMS Database 10.1.3 does certify with 10.x.

The corrected table is as follows:

Table 5–2   OEMS JMS Database Certification Matrix

OracleAS / Oracle Database	9.0.1	9.0.1.3	9.0.1.4	9.2.0.1	9.2.0.2+	10.1.0+	10.2.0+
9.0.2	X	X		X			
9.0.3			X			X	
9.04			X		X		
9.0.4.1						X	
10.1.2			X		X	X	
10.1.3			X		X	X	X

5.9.3.3 SQLServer Named Instance Definition Corrected

A SQL Server named instance defined outside of Java code, i.e., in `data-sources.xml` in OC4J, should have only one backward slash('\') separating the server name and the instance name.

This is incorrectly documented in the Data Sources chapter of the *Oracle Containers for J2EE Services Guide*.

Specifically, here's the change should be made to our documentation:

* Replace the text "myserver\\myinstance" with "myserver\myinstance"   in section "4.9.2.2.4 DataDirect SQLServer" of the Data Sources chapter in the OC4J Services

Specifically, replace the text "myserver\\myinstance" with "myserver\myinstance" in the following examples in the Data Sources chapter:

- SQLServer Managed Data Source

- SQLServer Native Data Source

5.9.3.4 Correction to ORMI Tunneling URL

In the "Configuring ORMI TunnelingThrough HTTP" section of Chapter 6, "Using Remote Invocation in OC4J", of the *Oracle Containers for J2EE Services Guide*, correct the following syntax specification:

Replace the INCORRECT syntax specification

```
ormi:http:// <hostname:[http_port]>/<context_uri>/<appName>
```

with the following CORRECT syntax specification:

```
ormi:http[s]://hostname[:http_port]>/<appName>
```

5.9.3.5 Missing Tables in Transaction Support Chapter

The tables are missing in the "OC4J Transaction Support Statistics" section of Chapter 5 - OC4J Transaction Support, of the 10.1.3.0 *Oracle Containers for J2EE Services Guide*.

This release note contains the text of the "OC4J Transaction Support Statistics" section with the tables restored.

OC4J Transaction Support Statistics

The statistics listed in the following table are provided by the JTAResource MBean, which is accessible through the Application Server Control Console.

Statistic	Description
activeCount	Total count of active transactions.
	A consistently high value can indicate a heavy load on the server.
	Suggest balancing across the cluster.
committedCount	Total count of transactions that have committed.
	This value can be used to determine the average load on the server since startup (or since the time the statistic was reset via admin). Similar to activeCount, a high value for this average can indicate a heavy load on the server.
	Suggest load balancing.
rolledbackCount	Total count of transactions that have rolled back.
	A high value might indicate an issue in the system (such as a database down) resulting in performance degradation.
	Examine the fine-grained rollback cause counts and logs for the cause of the rollbacks.
rolledbackDueToTimedOutCount	Total count of transactions that have rolled back due to timeout.
	A high number might indicate any of several issues causing the transaction (or activity within transactional bounds) to time out. Possibly the timeout value specified is not flexible enough.
	Determine what activities within the transactions involved (which may be of a certain type/application) are taking up time or adjust the transaction-timeout value in the transaction-manager.xml configuration file.

Statistic	Description
`rolledbackDueToAppCount`	Total count of transactions that have rolled back due to the application calling `setRollbackOnly` or rollback explicitly.
	A high value can occur for any reason, but will most often occur due to some handled exception within an application (such as SQLException during database update).
	Examine application code that calls `setRollbackOnly` or rollback to see why it is doing so.
`rolledbackDueToResourceCount`	Total count of transactions that have rolled back due to and error in an enlisted resource.
	A high value might indicate an issue with one or more resource managers (such as database) or with the network connection between OC4J and these resources.
	Examine the OC4J and resource manager logs.
`rolledbackDueToAdminCount`	Total count of transactions that have rolled back due to administrative action.
	A high value suggests that the system or application may not be automated enough (such as too much system administration in general or inadequate transaction architecture handling). Or possibly a particular issue has occurred that required extensive administration.
	Analyze logs for trends and contact those responsible for the administrative action.
`rollbackExceptionCount`	Total count of `RollbackExceptions` encountered.
	A high value might indicate an issue in the system (such as a database down) resulting in performance degradation. This might be caused by a direct internal system failure or by the application calling `setRollbackOnly` for some reason.
	Examine the fine-grained rollback cause counts and logs for the cause of the rollbacks and look into application code that calls `setRollbackOnly`.
`heuristicMixedExceptionCount`	Total count of `HeuristicMixedExceptions` encountered.
	A high value might indicate a high number of potentially non-ACID outcomes resulting from inconsistent or inappropriate administrative intervention.
	Analyze logs for trends and contact those responsible for the administrative action(s)

Statistic	Description
`heuristicRollbackExceptionCount`	Total count of Heuristic Rollback Exceptions encountered.
	A high value suggests that the system or application might not be automated enough (such as too much system administration in general or inadequate transaction architecture handling). Or possibly an issue has occurred that required extensive administration. Unlike the `rolledbackDueToAdminCount`, which indicates administrative rollback at the root transaction manager level while a transaction is active, this count indicates either a subordinate Transaction Manager or resource manager being administratively rolled back while in the prepared state.
	Analyze the OC4J and resource manager logs for trends and contact those responsible for the administrative action.
`securityExceptionCount`	Total count of `SecurityExceptions` encountered.
	A high value and most often any value greater than `0` might indicate an issue with the identity on the thread executing this.
	Examine the OC4J logs.
`illegalStateExceptionCount`	Total count of `IllegalStateExceptions` encountered.
	A high value here should be rare and only possible as a result of prior administrative intervention.
	Analyze OC4J logs for trends and contact those responsible for the administrative action.
`systemExceptionCount`	Total count of `SystemExceptions` encountered.
	A high value here should never occur and indicates a serious failure in the system.
	Analyze OC4J and resource manager logs.
`heuristicCommittedCount`	Total count of heuristically committed transactions.
	A high value suggests the system or application may not be automated enough (such as too much system administration in general or inadequate transaction architecture handling). Or possibly an issue has occurred that required extensive administration. This is due to a subordinate Transaction Manager and not a resource manager being administratively rolled back while in the prepared state.
	Analyze OC4J logs for trends and contact those responsible for the administrative action.

Statistic	Description
heuristicRolledbackCount	Total count of heuristically rolled back transactions.
	A high value suggests the system or application may not be automated enough (such as too much system administration in general or inadequate transaction architecture handling). Or possibly an issues has occurred that required extensive administration. This is due to a subordinate Transaction Manager and not a resource manager being administratively rolled back while in the prepared state.
	Analyze OC4J logs for trends and contact those responsible for the administrative action.
heuristicCount	Total count of all heuristically rolled back and committed transactions.
	See comments for heuristicCommittedCount and heuristicRolledbackCount.
averageCommitTime	Average commit time of all transactions.
	This is the average of the performTransaction values. However, this is a mean average so there may be spikes in the system indicating other issues as well.
	Analyze the finer-grain statistics as well as log files for indication of system spikes. Also analyze overall architecture as necessary.
performTransaction	Time from beginning to end of the transaction.
	This is useful as a high-level indicator of performance issues. However, since it is a measure from the beginning to the end of the transaction only, anything that occurs within this time frame might be the cause of a large value here. Some possibilities are: application logic, database activity, jms activity, transaction processing, and so on.
	Analyze finer-grain statistics and overall architecture as necessary.
singlePhaseCommitCompletion	Time required for a single-phase commit completion.
	A single-phase commit involves a single resource only and no 2PC costs (such as logging) are incurred. A large value here usually indicates an issue with the single resource being committed (such as network latency to the database).
	Analyze the metrics of the resource involved in the commit.
twoPhaseCommitCompletion	Time required for a two-phase commit completion.
	A high value indicates delays in the prepare and commit calls on resource managers or the transaction record logging in OC4J.
	Analyze the metrics of the resources involved in and performance settings for transaction record logging in OC4J in the transaction-manager.xml file.

Statistic	Description
transactionSuspended	Time a transaction has been suspended.
	A high value indicates that the transactions are being held in a suspended state waiting for a return call in a different or no transactional context (often from an EJB method call) or during propagation of a transaction context.
	Analyze the application to determine what activity is taking place during the suspend or whether there is a network latency in the case of propagation.
rollbackCompletion	Time required for a rollback completion.
	A high value indicates delays in the rollback calls on resource managers which may be a result of network latency or resource manager issues.
	Analyze the metrics of the resource(s) involved in the rollback.

The attributes listed in the following table are provided by the JTAResource MBean.

Attribute	Description
inDoubtXids	An array of in-doubt transaction Xids
activeXids	An array of active transaction Xids
heuristicCommittedXids	An array of heuristic committed transaction Xids
heuristicRolledbackXids	An array of heuristic rolled back transaction Xids
currentTransactionDetail	Details of all current transactions on the server, including those active, in-doubt, and recovering

The following JTAResource MBean operations are related to statistics.

Operation	Description
clearStats	Reset all OC4J Transaction Support statistics, except for activeCount, to 0
addThresholdEvent	Add an event to be fired when the specified OC4J Transaction Count Statistic exceeds the specified threshold and again at the specified count intervals
	addThresholdEvent takes the following parameters:
	■ statName - The name of the Count Statistic to monitor
	■ threshold1- The count at which to broadcast the event
	■ repeatNotificationInterval - The interval count at which to broadcast subsequent events

Operation	Description
`removeThresholdEvent`	Remove a threshold event
	`removeThresholdEvent` takes the following parameter:
	■ `statName` - The name of the Count Statistic whose threshold event is to be removed

5.9.3.6 Packager Flag to Prevent Generation of Web Services Web Module

If you want to prevent OC4JPackager from creating and adding a Web service module to your application, which would expose your XQS views with `WSDLvisibility` set to `"true"` as Web service operations, specify the `-no_ws` flag when you run the packager, as follows:

```
% java -jar OC4JPackager.jar ... -no_ws
```

This is an optional, toggle flag. By default, a Web service module will be added.

5.9.3.7 Wrong Data Source Element Name

The "Defining Data Source" chapter in the *OC4J Services Guide* contains an error. The Fifth bullet in the "Configuration Notes" section is incorrect. The bullet should read:

"Native data sources are defined using the `<native-data-source>` element. The `data-source-class` attribute can be set to any fully qualified class name of an object that implements the `javax.sql.DataSource` interface."

5.9.3.8 Wrong Defaults Listed for Connection Pool Attributes

The Data Source chapter in the *OC4J Services Guide* contains an error. The Connection Pool Attributes table incorrectly lists default values for the `time-to-live-timeout` attribute and the `abandoned-connection-timeout` attribute. The default value for these features is actually `0` and not `-1`. The default value (`0`) indicates that these features are disabled.

5.9.3.9 Invalid Attribute

Table 5-3, "Connection Pool Attributes," includes a description of the `disable-server-connection-pooling` attribute. This attribute is no longer supported and should not be added to the `J2EE_HOME/config/data-sources.xml` configuration file.

5.9.4 Oracle Containers for J2EE Developer's Guide

This section describes errors in the *Oracle Containers for J2EE Developer's Guide*. It covers the following item(s):

■ Section 5.9.4.1, "Import Example Only for Thin JDBC Drivers"

■ Section 5.9.4.2, "Default Session Timeout for OC4J Is 20 Minutes"

■ Section 5.9.4.3, "Correction to Example: Replacing the Oracle XML Parser with the Xerces Parser"

■ Section 5.9.4.4, "Missing Character in Service URL Strings"

■ Section 5.9.4.5, "Incorrect Setting for Client Module in orion-application.xml"

5.9.4.1 Import Example Only for Thin JDBC Drivers

In Chapter 3 of the *Oracle Containers for J2EE Developer's Guide*, "Utilizing the OC4J Class Loading Framework," the example of importing an earlier version of the Oracle JDBC driver is only for thin JDBC drivers. This example does not apply to the Oracle Call Interface (OCI) drivers.

You can find it in the subsection "Example: Importing an Earlier Version of the Oracle JDBC Driver" under "Configuring an Application to Import a Nondefault Version of a Shared Library." The example shows how to configure an application to use an Oracle 9.2.0_5 JDBC driver, which is an earlier version of the Oracle JDBC driver than the version packaged with Oracle Containers for J2EE 10*g* Release 3 (10.1.3).

5.9.4.2 Default Session Timeout for OC4J Is 20 Minutes

In the Oracle® Containers for J2EE Developer's Guide 10g Release 3 (10.1.3), Chapter 8, J2EE Best Practices, under the heading "Always Invalidate Sessions When No Longer in Use", the default session timeout for OC4J is incorrectly identified as 30 minutes.

The correct default session timeout for OC4J is 20 minutes.

5.9.4.3 Correction to Example: Replacing the Oracle XML Parser with the Xerces Parser

The following typo exists in the published version of the 10.1.3.0.0 *Oracle Containers for J2EE Developer's Guide*:

The sample on page 3-9, at the end of the subsection "Example: Replacing the Oracle XML Parser with the Xerces Parser" is missing a closing quotation mark in

```
max-version="2.5.0
```

The sample code line should read as follows:

```
<import-shared-library name="xerces.xml" max-version="2.5.0"/>
```

5.9.4.4 Missing Character in Service URL Strings

In *Oracle Containers for J2EE Developer's Guide*, Chapter 5, "Creating MBeans to Manage Your Applications," under "Remote Management Using the Management EJB (JSR-77)," some of the service URL strings in the code segments are missing a required colon character (":"). The incorrect entries are like this one:

```
String url="service:jmx:rmi///opmn://opmnhost1.company.com:6003/home"
```

In this entry, ":" is missing between "`rmi`" and "`///`".

The correct service URL string follows:

```
String url="service:jmx:rmi:///opmn://opmnhost1.company.com:6003/home"
```

5.9.4.5 Incorrect Setting for Client Module in orion-application.xml

The description of the `<client-module>` element recommends the wrong value for the `user` attribute under "Elements in the orion-application.xml File" in Appendix A, "OC4J-Specific Deployment Descriptors." The descriptions of the `auto-start` and `user` attributes incorrectly recommend setting `user` to `true` when `auto-start='true'`.

These attribute settings, however, would prevent OC4J from calling the main method in an application client archive (CAR) at startup. When the `auto-start` attribute is

set to `true`, you need to set `user` to `anonymous`. The descriptions of these `<client-module>` attributes should read as follows:

- `auto-start`: Whether to automatically start the application in-process at OC4J server startup. The default is `false`. If this attribute is set to `true`, the `user` attribute must be set to `anonymous`.

- `user`: Set to `anonymous` to run the client in-process. If the `auto-start` attribute is set to `true`, the `user` attribute must be set to `anonymous`.

5.9.5 Oracle Containers for J2EE Security Guide

This section describes errors in the *Oracle Containers for J2EE Security Guide*. It covers the following item(s):

5.9.5.1 Correct Package Name for getUserPrincipal() and getCallerPrincipal()

The OC4J Security Guide release for 10.1.3.0.0 specified an incorrect package name for the `User` class in its description of the `getUserPrincipal()` and `getCallerPrincipal()` methods. The package is `com.evermind.security`, not `oracle.j2ee.security`.

5.9.5.2 Obsolete <data-source> Element Cited

In the 10.1.3.1 and 10.1.3.0.0 *Oracle Containers for J2EE Security Guide*, in the section "Using Password Indirection" (chapter 6), a passage refers to the `password` attribute of the `<data-source>` element in file `data-sources.xml`. This actually should refer to the `password` attribute of the `<native-data-source>` element or `<managed-data-source>` element, which combined to replace the `<data-source>` element in the 10.1.3.0.0 release.

5.9.5.3 Form Authentication Method Supported with Custom Login Modules

The *Oracle Containers for J2EE Security Guide* incorrectly states in at least two locations that the form authentication method is not supported with custom security providers (custom login modules). This is incorrect—in 10.1.3.x implementations, the form method *is* supported with custom security providers.

The error occurs in the following locations:

- 10.1.3.0.0:

  - Chapter 2, "Overview of OC4J Security", section "Authentication in the OC4J Environment";

  - Chapter 13, "Web Application Security Configuration", section "Specifying auth-method in web.xml".

- 10.1.3.1 and higher:

  - Chapter 2, "Java Platform Security", section "Web Application Standard Authentication Methods"

– Chapter 17, "Web Application Security Configuration", section "Specifying auth-method in web.xml".

5.9.5.4 Invalid Reference to x509cert.mapping.attribute Property

The *Oracle Containers for J2EE Security Guide* incorrectly refers to the OracleAS JAAS provider property `x509cert.mapping.attribute`. In particular, Chapter 13, Web Application Security Configuration, in the section "Using Client-Cert Authentication" contains the following incorrect example:

```
<orion-application ... >
...
   <jazn provider="XML" ... default-realm="myrealm" ... >
      <property name="x509cert.mapping.attribute" value="CN"/>
...
   </jazn>
...
</orion-application>
```

This example and other references to `x509cert.mapping.attribute` are incorrect since the `x509LoginModule` utilizes the `mapping.attribute` property as do other providers. For more information about this property and how to configure it, see the release note at Section 5.1.29, "Specifying the Mapping Attribute"".

5.9.6 Oracle Containers for J2EE Administration and Configuration Guide

This section describes documentation errata for the *Oracle Containers for J2EE Configuration and Administration Guide*. It includes the following topics:

- Section 5.9.6.1, "Modification of the <log> Tag in system-application.xml for Log Rotation"

- Section 5.9.6.2, "Incorrect Example for Disabling Text File Logging"

- Section 5.9.6.3, "Format for Variables That Configure Text Logging for Web Sites"

- Section 5.9.6.4, "Configuration of OPMN Gateway Element Shows Incorrect Order"

- Section 5.9.6.5, "Starting Order Incorrect for Static Peer-to-Peer Replication"

5.9.6.1 Modification of the <log> Tag in system-application.xml for Log Rotation

In *Oracle Containers for J2EE Configuration and Administration Guide*, Chapter 1, "Introducing OC4J," under "Understanding the Application Hierarchy in OC4J" and "The System Application," the "Important" note states that you should not modify the `system-application.xml` file except for the <jazn> tag. You might also need to modify the <log> tag to rotate the system log file.

5.9.6.2 Incorrect Example for Disabling Text File Logging

In *Oracle Containers for J2EE Configuration and Administration Guide*, Chapter 10, "Logging in OC4J," under "Enabling/Disabling Text File Logging," the example for disabling text file logging comments out the entire `<log>` element, but the text before the example says to remove or comment out the `<file>` element. The incorrect example follows:

```
<!--
<log>
  <file path="application.log" />
</log>
```

```
-->
```

Here is a corrected example:

```
<log>
  <!-- <file path="application.log" /> -->
</log>
```

5.9.6.3 Format for Variables That Configure Text Logging for Web Sites

Information about the `$cookie` and `$header` variables is missing from the description of the `format` attribute of the `<access-log>` element under "Configuring Text-Based Access Logging" in Chapter 13, "Managing Web Sites in OC4J."

In the `<access-log>` subelement of the `<web-site>` element in a Web site's configuration file (`*-web-site.xml`), you can specify a number of variables in the `format` attribute, you can specify a number of variables that result in information being prepended to log entries. When you specify the `$cookie` or `$header` variable, the format needs to be as follows:

```
$cookie:[name]
$header:[name]
```

5.9.6.4 Configuration of OPMN Gateway Element Shows Incorrect Order

The order of `<topology>` subelements is incorrect in the `opmn.xml` configuration example under "Configuring Cross-Topology Gateways" in Chapter 8, "Configuring and Managing Clusters," of *Oracle Containers for J2EE Configuration and Administration Guide*. The example shows a configuration for a gateway as follows:

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

Because the `<topology>` subelements are incorrectly ordered according to the `opmn.xsd` file, using this configuration results in an OPMN error.

The correct configuration follows:

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

5.9.6.5 Starting Order Incorrect for Static Peer-to-Peer Replication

The following paragraph shows an incorrect starting order for nodes under "Configuring Static Peer-to-Peer Replication" in Chapter 9, "Application Clustering in OC4J," of *Oracle Containers for J2EE Configuration and Administration Guide*.

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

The following paragraph has the correct starting order.

In this configuration, each node specifies one other node as its peer. The result is that all of the nodes within the cluster are able to establish connections with one another. This scenario will work only if each node is started in succession; that is, `www3.company.com` must be started before `www2.company.com`. Otherwise, `www2.company.com` will not be able to "see" `www3.company.com`.

5.9.7 Oracle Containers for J2EE Deployment Guide

This section describes documentation errata for the *Oracle Containers for J2EE Deployment Guide*. It includes the following topics:

- Section 5.9.7.1, "Update References to oracle-ant.jar to Be ant-oracle.jar"

- Section 5.9.7.2, "Information Missing from Description of admin_client.jar -redeploy Command Subswitches"

- Section 5.9.7.3, "deploymentPlan Property Not Valid in Ant Task for WAR Deployment"

5.9.7.1 Update References to oracle-ant.jar to Be ant-oracle.jar

In *Oracle Containers for J2EE Deployment Guide*, Chapter 10, "Deploying with the OC4J Ant Tasks", under "Overview of the Ant Tasks", the `ant-oracle.jar` file is incorrectly referred to twice as the `oracle-ant.jar` file.

Correct the following sentences:

- Incorrect:

  "The `oracle-ant.jar` file is installed by default within the `ORACLE_HOME/ant/lib` directory."

  Correct:

  "The `ant-oracle.jar` file is installed by default within the `ORACLE_HOME/ant/lib` directory."

  The location is correct.

- Incorrect:

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

  Correct:

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

5.9.7.2 Information Missing from Description of admin_client.jar -redeploy Command Subswitches

Some information is missing from the description of the `-redeploy` command subswitches under "Redeploying an Archive" in Chapter 11, "Deploying with the admin_client.jar Utility":

- The `-file` subswitch can specify the name of an EAR, WAR, or RAR, not just an EAR.

- In the description of the `-sequential` subswitch, "EAR" should be "archive".

- The `-bindAllWebApps [webSiteName]` subswitch is missing. This optional subswitch binds all Web modules to the specified Web site, or to the `default` Web site if none is specified.

  You can optionally supply a value for `webSiteName`, which is the `name` portion of the `name_web-site.xml` file that configures the Web site.

5.9.7.3 deploymentPlan Property Not Valid in Ant Task for WAR Deployment

The `deploymentPlan` property is incorrectly listed in Table 10-6, "deploy Properties for Standalone WAR Deployment" in Chapter 10, "Deploying with the OC4J Ant Tasks." You cannot use a deployment plan to deploy a standalone Web module.

5.10 Oracle Application Server Containers for J2EE Job Scheduler

This section describes issues associated with Oracle Application Server Containers for J2EE Job Scheduler. It includes the following topics:

- Section 5.10.1, "Invalid Data Source Configuration May Result in Initialization Exception"

- Section 5.10.2, "Cancel API is not Transactional"

- Section 5.10.3, "Lower Than Expected Throughput may be Experienced for Large Number of Jobs"

- Section 5.10.4, "Removing a Job May Impact Job Scheduler Event Listener Processing"

- Section 5.10.5, "Peremptory Shutdown of OC4J Container may Prevent Subsequent Restart"

5.10.1 Invalid Data Source Configuration May Result in Initialization Exception

In the JDBC persistence configuration, a null pointer exception results on container startup if the associated data source is improperly configured or the database server is not up.

There is no workaround for this issue; make sure the data sources are configured correctly and the target database is up.

5.10.2 Cancel API is not Transactional

If the Cancel API is invoked within a JTA transaction, all outstanding executions are canceled synchronously, not after the transaction is committed.

There is no workaround for this issue.

5.10.3 Lower Than Expected Throughput may be Experienced for Large Number of Jobs

Lower than expected execution throughput may be observed when there are large burst jobs with concurrent schedules.

To work around this issue, disable the management bean and DMS statistics publication in order to increase throughput. This can be accomplished by setting the value of the following environment entries in the Job Scheduler configuration to false:

- `oracle.ias.scheduler.dms`
- `oracle.ias.scheduler.jmx`

5.10.4 Removing a Job May Impact Job Scheduler Event Listener Processing

Removing a job in the JMS persistence configuration may result in event processing delays. This behavior is exacerbated in a deployment where jobs are created and removed with high frequency.

To work around this issue, disable the management bean and DMS statistics publication in order to increase throughput. This can be accomplished by setting the value of the following environment entries in the Job Scheduler configuration to false:

- `oracle.ias.scheduler.dms`
- `oracle.ias.scheduler.jmx`

5.10.5 Peremptory Shutdown of OC4J Container may Prevent Subsequent Restart

The JMS server creates recover lock files in the *$ORACLE_HOME*`/j2ee/home/persistence` directory. As a result of a peremptory shutdown, these files may not be properly cleaned up and may prevent the container from restarting. Refer to the JMS release notes for more information.

This issue is only pertinent to Job Scheduler running in a JMS persistence environment.

6

Oracle Application Server Portal

This chapter describes issues associated with OracleAS Portal. It includes the following topic:

- Section 6.1, "General Issues and Workarounds"
- Section 6.2, "Portlet and Provider Issues and Workarounds"
- Section 6.3, "Documentation Errors"

6.1 General Issues and Workarounds

This section describes general issues and workarounds. It includes the following topic:

- Section 6.1.1, "Ensure Correct JDK Version in Compiler Settings"

6.1.1 Ensure Correct JDK Version in Compiler Settings

Java classes compiled using JDK 1.5 will not run in JDK 1.4. The default JDK version used to compile code depends on the JDK version used by the Oracle JDeveloper IDE.

If you create a portlet using Oracle JDeveloper 10*g* Release 3 (10.1.3) running on JDK 1.5, and plan to deploy it to an OC4J instance that uses JDK 1.4, then you must ensure that the compiler settings are set to 1.4. To set the project compiler settings, perform the following steps:

1. In Oracle JDeveloper, right-click your project in the Applications Navigator tab, and select **Project Properties** from the context menu.

2. From the Project Properties dialog box, click **Compiler** in the left pane.

3. From the **Source** list, select **1.4**.

4. Click **OK**.

6.2 Portlet and Provider Issues and Workarounds

This section describes issues and workarounds related to OmniPortlet, Web Clipping, Simple Parameter Form, Page portlet, and WSRP providers. This section includes the following topic:

- Section 6.2.1, "Error Configuring OC4J Standalone for OmniPortlet Deployment"

6.2.1 Error Configuring OC4J Standalone for OmniPortlet Deployment

As part of the procedure to deploy OmniPortlet in a multiple middle-tier environment, you are required to create a database preference store to store the OmniPortlet preference information.

If you installed OC4J release 10.1.3, and added a `data-source` entry in the `data-sources.xml` file, then you will get errors. This is because data sources are defined differently in OC4J release 10.1.3 due to a change in the format of the `data-sources.xml` file.

To configure OC4J release 10.1.3 for OmniPortlet deployment, perform the following steps:

1. Add a new entry in the `data-sources.xml` file as described in step 8 under Section 6.3.2, "Configuring OC4J Standalone" in *Oracle Application Server Portal Developer's Guide*.

2. Edit the `provider.xml` file located in the directory, *ORACLE_ HOME*`/j2ee/OC4J_ Portal/applications/portalTools/omniPortlet/WEB-INF/providers /omniPortlet`. Edit the `preferenceStore` tag as shown in bold:

```
<provider class="oracle.webdb.reformlet.ReformletProvider">
    <vaultId>0</vaultId>
    <session>true</session>
    <preferenceStore
class="oracle.portal.provider.v2.preference.DBPreferenceStore">
        <name>omniPortletprefStore</name>
        <connection>jdbc/PooledConnection</connection>
    </preferenceStore>
```

3. Restart OC4J.

6.3 Documentation Errors

This section describes known errors in OracleAS Portal documentation. It includes the following topic:

■ Section 6.3.1, "Edit Defaults Mode"

6.3.1 Edit Defaults Mode

In *Oracle Application Server Portal Developer's Guide*, the last list item under Section 3.2.6, "Edit Defaults Mode" reads as follows:

Configure OmniPortlet to use the DBPreferenceStore, and follow the steps in Section "5.3.6 Step 6: Configure Portal Tools and Web Providers (Optional)" of *Oracle Application Server Portal Configuration Guide*.

These steps in *Oracle Application Server Portal Configuration Guide* are applicable *only* for OC4J versions prior to 10.1.3. This list item should read as follows:

Configure OmniPortlet to use the database preference store. Depending on the version of OC4J that you have installed, perform either of the following:

■ If you have installed an OC4J release prior to 10.1.3, then perform the steps described in section "5.3.6 Step 6: Configure Portal Tools and Web Providers (Optional)" of Oracle Application Server Portal Configuration Guide.

■ If you installed OC4J release 10.1.3, then perform the following steps:

1. Add a new entry in the `data-sources.xml` file as described in step 8 under Section 6.3.2, "Configuring OC4J Standalone" in *Oracle Application Server Portal Developer's Guide*.

2. Edit the `provider.xml` file located in the directory, *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet`. Edit the `preferenceStore` tag as shown in bold:

```
<provider class="oracle.webdb.reformlet.ReformletProvider">
    <vaultId>0</vaultId>
    <session>true</session>
    <preferenceStore
class="oracle.portal.provider.v2.preference.DBPreferenceStore">
        <name>omniPortletprefStore</name>
        <connection>jdbc/PooledConnection</connection>
    </preferenceStore>
```

3. Restart OC4J.

7

Oracle TopLink

This chapter describes issues associated with Oracle TopLink (TopLink). It includes the following topic:

- Section 7.1, "General Issues and Workarounds"
- Section 7.2, "Documentation Errata"

7.1 General Issues and Workarounds

This section describes general issue and workaround. It includes the following topic:

- Section 7.1.1, "Links to Hosted Documentation and Web-Based Resources from TopLink Workbench"
- Section 7.1.2, "Using Non-ASCII Characters with a JAXB 1.0 TopLink Project"
- Section 7.1.3, "TopLink Workbench Look and Feel With Linux GTK"
- Section 7.1.4, "Unit of Work and JTA Transactions"
- Section 7.1.5, "UnitOfWork.release() not Supported with External Transaction Control"
- Section 7.1.6, "ReportQuery Results for Aggregate Functions may be Truncated"
- Section 7.1.7, "Attribute Joining of One-to-One Mappings not Supported with Inheritance"
- Section 7.1.8, "Using Oracle Database Advanced Data Types may Fail with Some Data Sources"
- Section 7.1.9, "Returning Policy with Optimistic Locking"
- Section 7.1.10, "Using Timestamp"
- Section 7.1.11, "Configuring Sequencing in sessions.xml"
- Section 7.1.12, "Marshalling a Non-root Object with Document Preservation"

7.1.1 Links to Hosted Documentation and Web-Based Resources from TopLink Workbench

When using the online help and the hosted documentation, TopLink Workbench requires an internet connection to OTN (Oracle Technology Network). Your PC must have a network connection (including any required proxy information) and an internet browser for use with the TopLink Workbench. If the TopLink Workbench is unable to connect to OTN, some links in the online help or the Welcome page may not work properly.

To configure preferences for an external browser for use with the TopLink Workbench, select **Tools** > **Preferences** > **Help**. Click **Browse** and select the location of your default Web browser. For more information on Help preferences, go to the Help menu and select *Working With TopLink Workbench Preferences*.

7.1.2 Using Non-ASCII Characters with a JAXB 1.0 TopLink Project

When you generate class and method names that include non-ASCII characters, TopLink will throw an exception. This problem occurs when creating a JAXB 1.0 TopLink project from an XML schema that contains non-ASCII characters. Ensure that your XML schema does not contain any non-ASCII characters.

7.1.3 TopLink Workbench Look and Feel With Linux GTK

When using TopLink Workbench on a PC using the Linux GTK look and feel, do not set the TopLink Workbench Look and Feel Preference to **GTK+**. Doing so may cause unexpected complications. Oracle recommends using the **Metal**, **Oracle**, or **CD/Motif** look and feel instead. Refer to "Working with TopLink Workbench Preferences" in *Oracle TopLink Developer's Guide* for complete information.

7.1.4 Unit of Work and JTA Transactions

If a unit of work is synchronized with a Java Transaction API (JTA), an exception will be thrown if it is released. If the current transaction requires that changes not be persisted, the JTA transaction must be rolled back. If you wish to have the transaction rolled back, do so with its API.

7.1.5 UnitOfWork.release() not Supported with External Transaction Control

A unit of work synchronized with a Java Transaction API (JTA) will throw an exception if it is released. If the current transaction requires its changes to not be persisted, the JTA transaction must be rolled back.

When in a container-demarcated transaction, call `setRollbackOnly()` on the EJB/session context:

```
@Stateless
public class MySessionBean
{
    @Resource
    SessionContext sc;

    public void someMethod()
    {
        ...
        sc.setRollbackOnly();
    }
}
```

When in a bean-demarcated transaction then you call `rollback()` on the UserTransaction obtained from the EJB/session context:

```
@Stateless
@TransactionManagement(TransactionManagementType.BEAN)
public class MySessionBean implements SomeInterface
{
    @Resource
    SessionContext sc;
```

```
public void someMethod()
{
    sc.getUserTransaction().begin();
    ...
    sc.getUserTransaction().rollback();
}
}
```

7.1.6 ReportQuery Results for Aggregate Functions may be Truncated

When using a ReportQuery to return calculated values, the data type of the column is used to convert the returned results. As a result, values returned for mapped attributes using average, variance, and standard deviation will be converted into the mapped attribute type and could result in loss of precision through truncation.

To work around this issue, use the field instead of the attribute's query key. For example:

- Using the attribute's query key results in truncated value:

  ```
  rq.addAverage("salary");
  ```

- Use the database column instead to avoid truncation:

  ```
  rq.addAverage("salary", eb.getField("SALARY.SALARY"));
  ```

7.1.7 Attribute Joining of One-to-One Mappings not Supported with Inheritance

It is not possible to execute a query on a class involved in an inheritance hierarchy which also has one-to-one joined attributes configured. This issue has already been addressed in TopLink Essentials and the solution will be available in the next release or a patch-set of this release.

7.1.8 Using Oracle Database Advanced Data Types may Fail with Some Data Sources

When using Oracle9i and Oracle10g database platform-provided advanced data types with some data source implementations, failures may occur. The platform does not or cannot correctly unwrap the raw Oracle specific JDBC connection and cannot complete the necessary conversions. To avoid this issue, you can use TopLink internal connection pooling.

7.1.9 Returning Policy with Optimistic Locking

The returning policy, which allows values modified during INSERTs and UPDATEs to be returned and populated in cached objects, does not work in conjunction with numeric version optimistic locking. The value returned for all UPDATEs is **1** and does not provide meaningful locking protection.

Do not use numeric optimistic locking in conjunction with a returning policy.

7.1.10 Using Timestamp

TopLink assumes that date and time information returned from the server will use `Timestamp`. If the JDBC driver returns a `String` for the current date, TopLink will throw an exception. You should use a driver that returns `Timestamp` or change the policy to use local time instead of server time.

7.1.11 Configuring Sequencing in sessions.xml

When configuring a custom sequence table for a session, TopLink Workbench will throw an exception and your edits will not be written to the `sessions.xml` when it is saved. As a workaround, configure sequencing at the project level. See "Configuring Sequencing at the Project Level" in the *Oracle TopLink Developer's Guide* for details.

7.1.12 Marshalling a Non-root Object with Document Preservation

When using document preservation (see "Configuring Document Preservation" in the *Oracle TopLink Developer's Guide*), only a root object can be marshaled. Attempting to marshal a non-root object with document preservation on will result in an exception.

7.2 Documentation Errata

This section includes information on the following issue:

-

7.2.1 Creating project.xml Programatically

The command for creating the project.xml file programatically, as listed in the *Oracle TopLink Developer's Guide*, is incorrect. The correct command is:

```
java -classpath
toplink.jar;toplinkmw.jar;xmlparserv2.jar;ejb.jar;oracle.toplink.workbench.mapping
s.DeploymentXMLGenerator <MW_Project.mwp> <output file.xml>
```

8

Oracle Business Rules

This chapter describes issues associated with Oracle Business Rules. It includes the following topics:

-

8.1 Using RL Reserved Words in Java Package Names

Invalid RL Language is generated if an RL Language reserved word (for example, the word `rule` in `mypkg.rule.com`) is part of the Java package name. If an RL Language reserved word is used in a Java package name, an error message like the following appears:

```
Oracle RL 1.0: syntax error PareseException: encountered 'rule' when expecting
one of: <XML_IDENTIFIER> ...<IDENTIFIER> ... "*" at line 11 column 19 in main
```

There is no workaround for this issue; do not use RL Language reserved words in Java package names.

8.2 Ancestor Methods are not Visible from Sub-Classes

The properties of a superclass are visible in the appropriate choice lists, but the methods of the ancestor classes are not visible.

There is no workaround for this issue.

8.3 New and Deleted Patterns are not Immediately Available

When you use Rule Author to add a pattern to a rule, the list of actions does not immediately reflect the existence of the new pattern instance. Similarly, when you use Rule Author to delete a pattern from a rule, the list of actions does not immediately reflect this change.

To work around this issue, you can do either of the following:

1. Update the rule by clicking either **OK** or **Apply**, then re-select the rule from the navigation pane.

2. Create a new action.

8.4 Changes to the Bind Variable Name are not Immediately Reflected

If you change the bind variable name on the Pattern Definition page, this change is not immediately reflected on the parent page (the Rule Author Rule page). In order to see your changes, you must close the Pattern Definition page before continuing your work on the Rule Author Rule page.

There is no workaround for this issue.

8.5 Refreshing Imported Java Facts After Changing the Underlying Classes

To refresh imported Java Facts after the underlying classes are changed, re-import the Java classes. The JavaFact is updated with the new methods and properties.

If you want to change the property name in the Java class, you must recreate any rules that use the property whose name you want to change.

If you want to change the property type or visibility, you must:

1. Remove any references to that property in the rules (but not necessarily the fact type itself).

2. Remove the property entirely from the Java class.

3. Re-import the Java class without the property.

4. Add the property back to the class with the new type or visibility.

5. Re-import the Java class again.

Methods can always be re-imported regardless of whether or not the return type is changed. However, if the return type is changed, any rule patterns that use this function will be invalid. However, you will not see that they are invalid until you attempt to validate or save the rule, so you should exercise caution when changing the return type.

8.6 RL serviceDateValidation Error: Fact Class Should Be Declared Earlier

In certain cases, you can receive the following RL generation error:

"Fact class should be declared earlier ...".

For example, with the following generated RL for a data model:

```
ruleset DM {
 function CreateDataAndTestRules()
 {
 servOrders.ServiceOrderTypeImpl sot = new
servOrders.ServiceOrderTypeImpl();
.
.
.
```

This causes the implemented `Impl` class to be autofactclassed before the Type is factclassed. When the Type is then referred to in the rules, the following error is thrown:

ClassName: A syntax error is found. Error:fact class should be declared earlier at line 5 column 3 in `serviceDateValidation`

To fix this problem:

Classes with complex type hierarchies, for example they both extends a class and implements an interface must have classes referenced in a particular order. If the classes are not referenced in the correct order, then the rules engine cannot correctly determine the class hierarchy.

There are two ways to fix this problem:

- Check "supports XPath" for the `ServiceOrderType` class; this introduces a slight performance cost.

- Change the code in the `CreateDataAndTestRules` function. For example, change the instance creation to:

    ```
    servOrders.ServiceOrderType sot = new servOrders.ServiceOrderTypeImpl();
    ```

    This parses in the right order such that the superclass is created before the implementing class.

9

OracleAS Disaster Recovery

This chapter describes issues associated with OracleAS Disaster Recovery. It includes the following topics:

- Section 9.1, "General Issues and Workarounds"

9.1 General Issues and Workarounds

This section describes general issues and workarounds. It includes the following topic:

- Section 9.1.1, "Compatible ASG Releases for OracleAS Instances from Different Oracle Application Server Releases"

- Section 9.1.2, "Compatible ASG Releases in an OracleAS Disaster Recovery Topology"

- Section 9.1.3, "Adding an Instance from a Remote Client Adds an Instance on the Local Instance and Not on the Remote Instance"

- Section 9.1.4, "Switchover Operation in an Asymmetric Topology Requires All Components to be Shutdown on Instances on the Primary Site that Do Not Have a Standby Peer"

- Section 9.1.5, "ASG to Catch Array Overflow Exceptions in Queries to Primary"

- Section 9.1.6, "The create standby Command Fails if the Redo Log Files Directories do not Exist at the Standby"

- Section 9.1.7, "Corrupt Index Blocks in Metadata Repository Databases"

9.1.1 Compatible ASG Releases for OracleAS Instances from Different Oracle Application Server Releases

By default, when you install an Oracle Application Server instance using a particular release of Oracle Application Server, a particular release of Application Server Guard (ASG) is installed into the Oracle home for the instance. You also install ASG on standalone hosts on which external resources (such as an Oracle database) are located that you have included in your OracleAS Disaster Recovery topology.

Multiple releases of Application Server Guard (ASG) are available. It is possible (recommended) in some cases to upgrade the ASG release that was installed in an Oracle Application Server instance home when you installed that instance. To upgrade the ASG release in an Oracle Application Server instance home, download the ASG standalone kit for the recommended ASG release from the Oracle Technology Network (OTN), and then use that ASG standalone kit to install the recommended ASG release into the home. You also use the ASG standalone kit to install ASG on standalone hosts that are included in your OracleAS Disaster Recovery topology.

Use Table 9–1 and Table 9–2 to determine whether a particular ASG release is compatible when installed into an Application Server instance home for a particular Oracle Application Server release. The left column of the table shows the different ASG releases for which a standalone ASG installation kit is available. The remaining columns show different Oracle Application Server releases for which an Oracle Application Server instance can be created.

This list describes the meaning of the entries in Table 9–1 and Table 9–2:

- **N**: This ASG release is not compatible with an instance from this Oracle Application Server release.

- **X**: This ASG release cannot be installed into the Oracle home for an instance from this Oracle Application Server release.

- **Y-NR**: This ASG release is compatible with an instance from this Oracle Application Server release, but Oracle recommends that you *do not* install this ASG release into the instance's Oracle home because another ASG release is recommended.

- **Y**: This ASG release is compatible with an instance from this Oracle Application Server release. Oracle recommends you install this ASG release into the instance's Oracle home.

Table 9–1 shows the compatible ASG releases for Oracle Application Server instances from Oracle Application Server 10.1.2.0.2 through 10.1.3.3.

Table 9–1   Compatible ASG Releases for OracleAS Instances from Releases 10.1.2.0.2 Through 10.1.3.3

ASG Release	10.1.2.0.2 OracleAS Instance	10.1.2.1 OracleAS Instance	10.1.2.2 OracleAS Instance	10.1.3.0 OracleAS Instance	10.1.3.1 OracleAS Instance	10.1.3.2 OracleAS Instance	10.1.3.3 OracleAS Instance
10.1.2.0.2	Y-NR	X	X	N	N	N	N
10.1.2.2	Y	Y	Y	N	N	N	N
10.1.2.2.1 (ASG-only release)[1]	Y	Y	Y	N	N	N	N
10.1.3.0	N	N	N	Y-NR	X	N	X
10.1.3.1	N	N	N	Y-NR	Y-NR	Y-NR	X
10.1.3.3	N	N	N	Y	Y	Y	Y

[1] This is the ASG release that was provided (installed by default) with the OracleAS 10.1.4.2 release. It is compatible with the OracleAS 10.1.2.x releases. There is no OracleAS 10.1.2.2.1 release.

For example, if you have an Oracle Application Server 10.1.3.1 instance and you want to know which ASG release to install in the instance home, you can use Table 9–1 to determine the following:

- No ASG 10.1.2.x release is compatible with an Oracle Application Server 10.1.3.1 instance.

- The ASG 10.1.3.0 release cannot be installed into the Oracle home for an Oracle Application Server 10.1.3.1 instance.

- The ASG 10.1.3.1 release is compatible with an Oracle Application Server 10.1.3.1 instance, but Oracle recommends that you *do not* install the ASG 10.1.3.1 release into the Oracle home for an Oracle Application Server 10.1.3.1 instance.

■ The ASG 10.1.3.3 release is compatible with an Oracle Application Server 10.1.3.1 instance and Oracle recommends that you install the ASG 10.1.3.3 release into the Oracle home for an Oracle Application Server 10.1.3.1 instance.

Table 9–2 shows the compatible ASG releases for Oracle Application Server instances from Oracle Application Server 10.1.4.0 through 10.1.4.2.

Table 9–2   Compatible ASG Releases for OracleAS Instances from Releases 10.1.4.0 Through 10.1.4.2

ASG Release	10.1.4.0 OracleAS Instance[1]	10.1.4.1 OracleAS Instance[2]	10.1.4.2 OracleAS Instance[3]
10.1.2.0.2	Y-NR	Y-NR	X
10.1.2.2	Y-NR	Y-NR	Y-NR
10.1.2.2.1 (ASG-only release)[4]	Y	Y	Y
10.1.3.0	N	N	N
10.1.3.1	N	N	N
10.1.3.3	N	N	N

[1] ASG 10.1.2.0.2 is installed by default.

[2] ASG 10.1.2.0.2 is installed by default.

[3] ASG 10.1.2.2.1 is installed by default.

[4] This is the ASG release that was provided (installed by default) with the OracleAS 10.1.4.2 release. It is compatible with the OracleAS 10.1.2.x releases. There is no OracleAS 10.1.2.2.1 release.

9.1.2 Compatible ASG Releases in an OracleAS Disaster Recovery Topology

This chart shows which ASG release combinations are compatible in in an OracleAS Disaster Recovery topology. A topology is a collection of Oracle Application Server instance homes and standalone host homes that combine to comprise the OracleAS Disaster Recovery production site and standby site. Each Oracle Application Server home has a specific ASG release installed, either by default or by a standalone ASG installation. Since OracleAS Disaster Recovery operations are distributed, there is a collection of recommended ASG releases that should be installed across the topology. Example for the EDG deployment, the collection of Oracle homes should be upgraded to ASG version 10.1.3.3 or 10.1.2.2.x.

Use Table 9–3 to determine whether two ASG releases are compatible in an OracleAS Disaster Recovery topology. Find the first ASG release in the left column of the table and then find the second ASG release in one of the other columns of the table.

This list describes the meaning of the entries in Table 9–3:

■ **Y-NR**: The first ASG release is compatible with the second ASG release, but Oracle recommends that you *do not* use this ASG release combination in your topology.

■ **Y**: The first ASG release is compatible with the second ASG release. Oracle recommends that you use this ASG release combination in your topology.

Table 9–3 shows which ASG releases are compatible with other ASG releases.

Table 9–3   Compatible ASG Releases in a Topology

ASG Release	10.1.2.0.2	10.1.2.2	10.1.2.2.1	10.1.3.0	10.1.3.1	10.1.3.3
10.1.2.0.2	Y-NR	Y-NR	Y-NR	Y-NR	Y-NR	Y-NR
10.1.2.2	Y-NR	Y	Y	Y-NR	Y-NR	Y
10.1.2.2.1	Y-NR	Y	Y	Y-NR	Y-NR	Y
10.1.3.0	Y-NR	Y-NR	Y-NR	Y-NR	Y-NR	Y-NR
10.1.3.1	Y-NR	Y-NR	Y-NR	Y-NR	Y-NR	Y-NR
10.1.3.3	Y-NR	Y	Y	Y-NR	Y-NR	Y

9.1.3 Adding an Instance from a Remote Client Adds an Instance on the Local Instance and Not on the Remote Instance

When using the asgctl `add instance` command, the OracleAS Guard client must be run from a system that is already included in the topology.

For example, when an OracleAS Guard client is connected to the OracleAS Guard server that is to be added to an existing topology, the following error is returned:

```
ASG_IAS-15785: ERROR: The topology is missing the instance that exists in the home
where the ASG server is running.
You must first discover or add the instance in home
```

The workaround to this problem is to use an OracleAS Guard client from a system that is already included in the topology to perform the asgctl `add instance` command to add an instance to the topology.

9.1.4 Switchover Operation in an Asymmetric Topology Requires All Components to be Shutdown on Instances on the Primary Site that Do Not Have a Standby Peer

Prior to performing an asgctl switchover operation in an asymmetric topology for instances that do not have a standby peer, you must perform an opmnctl stopall command to shutdown all components on each of these ignored instances on the primary site.

When an XML policy file is in use for an asymmetric topology and has the <instanceList successRequirement ="Ignore" set for an instance, for example, as shown in the following example, then in a switchover operation OracleAS Guard ignores that instance:

```
.
.
.
<instanceList successRequirement = "Ignore">
  <instance>instance B</instance>
</instanceList>
.
.
.
```

OracleAS Guard, on a switchover operation, shuts down all components on the old primary site except for OracleAS Guard and OPMN and ignores instance B because the policy file specifies to do so. The switchover operation fails because all components are not shut down on the primary site, in this case instance B because the policy file specifies to ignore instance B on the primary site, which has no standby peer.

To workaround this problem, the OracleAS Disaster Recovery Administrator must perform an opmnctl stopall operation for all components on instance B prior to the switchover operation in order for the switchover operation to succeed in this asymmetric topology.

9.1.5 ASG to Catch Array Overflow Exceptions in Queries to Primary

The asgctl `create standby database` command may fail with an ASG_DUF-4950 error and an error message stack on the console similar to the following:

```
ASG_DUF-4950: An error occurred on host "myhost" with IP "1.1.1.1" and port "7890"
ASG_SYSTEM-100: 10
ASG_DUF-4900: An exception occurred on the server.
ASG_DGA-13009: Error during Create Physical Standby
```

In the duf_client.log file on the primary site DB node, the following messages would be found:

```
java.lang.ArrayIndexOutOfBoundsException: 10
at oracle.duf.DufJdbc.queryRedoLogInfo(DufJdbc.java:535)
at oracle.duf.DufDb$jdbc.queryRedoLogInfo(DufDb.java:4966)
at oracle.duf.DufDb$jdbc.access$2000(DufDb.java:4884)
at oracle.duf.DufDb.queryRedoLogInfo(DufDb.java:3439)
```

This problem occurs because ASG does not properly handle a database configured with 10 or more redo log files.

To avoid this problem, reduce the number of database redo log files to fewer than 10.

This problem is fixed in release 10.1.3.3.

9.1.6 The create standby Command Fails if the Redo Log Files Directories do not Exist at the Standby

The asgctl `create standby database` command fails if the redo log files directories on the production site do not exist at the standby.

ASG expects the target redo log directory structure to be symmetrical with the redo log directory structure on the production site. If they do not exist in the same directories, a failure will result with the following output:

```
ASG_ORACLE-300: ORA-00301: error in adding log file '/PATH/redo010.log' - file
cannot be created
ASG_ORACLE-300: ORA-27040: file create error, unable to create file
ASG_DUF-3700: Failed in SQL*Plus executing SQL statement:  ALTER DATABASE ADD
STANDBY LOGFILE GROUP 10
'/PATH/redo010.log' SIZE 52428800 /* ASG_DGA */;.
ASG_DUF-3535: Failed to create standby redo log.
ASG_DUF-3535: Failed to create standby redo log.
ASG_DGA-13011: Error during Create Physical Standby: Finish-configure standby.
ASG_DUF-3027: Error while executing Creating physical standby database - finish
phase at step - finish step.
```

To avoid this problem, create the target redo log file directories on the standby site in the same directories as the production site.

9.1.7 Corrupt Index Blocks in Metadata Repository Databases

After an ASG switchover or failover operation, metadata index block corruption may occur in a Disaster Recovery metadata repository database.

For a full description of the problem and an explanation of how to deal with it, refer to Oracle*MetLink* note 386830.1 at:

http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=386830.1

10

Oracle Sensor Edge Server

This chapter describes issues with Oracle Sensor Edge Server. It includes the following topics:

10.1 Installation and Configuration Issues

This section describes installation and configuration issues regarding Oracle Sensor Edge Server. It includes the following topics:

10.1.1 Valid 10.1.3 OC4J in Correct Oracle Home Required for Oracle Sensor Edge Mobile

You must have OC4J 10.1.3 installed into the same Oracle home into which you are going to install Oracle Sensor Edge Mobile. Having OC4J in another location is not supported by Oracle Universal Installer.

After confirming that OC4J is installed in the proper place, you can install Oracle Sensor Edge Server products. To install any Oracle Sensor Edge Server products, you must first select Oracle Sensor Edge Server in the Installation screen. Subsequently, you will be able to install Oracle Sensor Edge Server, Oracle Sensor Edge Mobile, or both products.

If you install both products, you will be given the option of launching the Sensor Edge Welcome twice. Both products are trying to open the same Welcome page; opening either one opens the same page.

10.1.2 OC4J 10.1.2 Must be Stopped if Installing Against It

As noted in Oracle Sensor Edge Server Guide, you can install Oracle Sensor Edge Server along with OC4J 10.1.2 (versus the current 10.1.3 version). If you choose to

install Oracle Sensor Edge Server with OC4J 10.1.2, ensure that OC4J is not running at the time. If OC4J 10.1.2 is running when you try to install Oracle Sensor Edge Server against it, the installation will fail.

10.1.3 Default Database Tablespaces

The scripts included with Oracle Sensor Edge Server are for a simple, generic installations. Our testing is accomplished against such generic installations.

However, in enterprise environments, there may be many custom requirements and configurations that customers must design and implement themselves.

For this reason, please ensure that you are familiar with tablespace creation. To learn about database tablespace creation, see the *Oracle SQL Reference* and *Oracle Database Concepts* guides for your particular release.

Once you have set up your database (including the server, datafiles, and tablespaces across datafiles), you can modify Oracle Sensor Edge Server install scripts as needed.

`create_edg_user.sql` is used for SDS, and `create_edg_sda_user.sql` is used for SDR.

These scripts must be modified to use newly-created tablespaces for specific customer requirements.

Under the create user command where you specify which tablespace the user is to use, change the line containing 'create user' in the .sql scripts, according to the manual.

10.1.4 Oracle Sensor Edge Server Installation Fails

When installing the Oracle Sensor Edge Server, do not select the Oracle Containers for J2EE 10.1.3.0.0 option from the companion CD Oracle Universal Installer (OUI); selecting this option along with the SES option causes the Oracle Sensor Edge Server installation to fail. The valid OC4J instance is installed using the Oracle Application Server CD-ROM (or DVD-ROM), not the companion CD OUI. For more information, see the Oracle Application Server Installation Guide appropriate to your platform.

10.1.5 Resolving the Reboot Issue for XML Configuration

Error messages regarding invalid jms.xml typically occur because of an abnormal termination of OC4J, an OC4J crash, or the IP address of the server running OC4J changes.

If you encounter OC4J JMS Server startup problems after an abnormal shutdown, first check that no other OC4J JMS Server is running and using the same persistence files. Remove any .lock files from the ORACLE_HOME/j2ee/instance_name/persistence directory and then try restarting again.

If problems persist, confirm that the jms.xml file is valid.

If problems still persist, remove the jms.state file from the persistence directory and try again. Removing this file may result in the loss of transaction information. See also the section entitled "Abnormal Termination", located in the "Resource Providers" in "Chapter 3: Oracle Enterprise Messaging Service (OEMS)" of *Oracle Containers for J2EE Services Guide*.

10.2 General Issues

This section describes general issues encountered in Oracle Sensor Edge Server. It includes the following topics:

- Section 10.2.1, "Using UTL_EDG.REMOVE_RULE Displays an Error"
- Section 10.2.2, "Adding a Rule Displays an Error"
- Section 10.2.3, "Localization -- Navigation Tree in the SES Console Renders as the Server Locale-Defined Character Set"
- Section 10.2.4, "Reassignment of Audio Event Type (207)"

10.2.1 Using UTL_EDG.REMOVE_RULE Displays an Error

When using the API `UTL_EDG.REMOVE_RULE`, an error may be encountered. This API only works if your rule was created through other Oracle Sensor Edge Server APIs.

10.2.2 Adding a Rule Displays an Error

In this release, be sure to use the syntax `':event'` instead of `'TAB.USER_DATA'`.

10.2.3 Localization -- Navigation Tree in the SES Console Renders as the Server Locale-Defined Character Set

The character set defined in the Server Locale overrides the characters set in the browser locale even when you switch the browser locale. For example, if you switch the browser locale to *Japanese*, the Server Locale is defined as traditional Chinese, the strings in the Navigation Tree of the SES Console (such as the *Available Extensions*, filters, devices, and device group nodes) and the error and confirmation messages render in traditional Chinese rather than in Japanese. To correct this problem, reset the Server Locale, or override the Server Locale-defined character set by overloading *MsgTranslator* methods to take locale. String literals in the SESConfig tree control must be non-static and fetched using *Locale* on each page request.

10.2.4 Reassignment of Audio Event Type (207)

The Audio Event Type (which is the event supported by the Simple Audio Driver) has been reassigned from *207* to *104*, and is now included in the group Generic Instructions to Devices (Event Types 100 - 199). Its Subtype remains *1* (*Play audio jobs in the .xml file in the Datafield*).

10.3 Documentation Issues

This section describes documentation issues regarding Oracle Sensor Edge Server. It includes the following topic:

- Section 10.3.1, "Documentation for Oracle Sensor Edge Server Extensions"
- Section 10.3.2, "Manually Deploying Sensor Data Streams Against an Existing Sensor Data Repository"

10.3.1 Documentation for Oracle Sensor Edge Server Extensions

Once the server is installed, you can see documentation for extensions at:
http://<host on which SES is installed>:<oc4j

`port>/edge/extensions`. Or, see Oracle Sensor Edge Server information on Oracle
Technology Network at:
`http://www.oracle.com/technology/products/sensor_edge_`
`server/extensions.html`.

10.3.2 Manually Deploying Sensor Data Streams Against an Existing Sensor Data Repository

Oracle Sensor Edge Server Guide omits a step in describing how to manually deploy
Sensor Data Streams if you have already created a Sensor Data Repository. The
procedure for manually deploying Sensor Data Streams if the Sensor Data Repository
exists is as follows:

1. Use SQL*Plus to connect to the database as *sysdba*, by running `sqlplus /nolog`.

2. Run sqlplus; connect as *sys/your_pwd@your_db as sysdba*.

3. Run the script `grant_edg_user.sql`, located in `Oracle_Home`
   `/edge/stage/sql/10.1.3`.

4. Disconnect as *sys* and then reconnect to the database as the *edge user*.

5. Run the script `edg_create_streams.sql`, located in `Oracle_Home`
   `/edge/stage/sql/10.1.3`.

6. Follow the procedure described in "Connecting Oracle Sensor Edge Server to
   Sensor Data Streams".